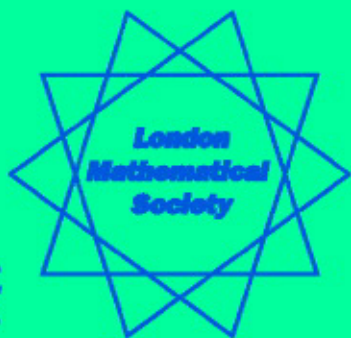


London Mathematical Society
Lecture Note Series 317

Advances in Elliptic Curve Cryptography

Edited by
Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart

CAMBRIDGE
UNIVERSITY PRESS



This page intentionally left blank

Managing Editor: Professor N.J. Hitchin, Mathematical Institute,
University of Oxford, 24–29 St Giles, Oxford OX1 3LB, United Kingdom

The titles below are available from booksellers, or from Cambridge University Press at www.cambridge.org

- 152 Oligomorphic permutation groups, P. CAMERON
- 153 L-functions and arithmetic, J. COATES & M.J. TAYLOR (eds)
- 155 Classification theories of polarized varieties, TAKAO FUJITA
- 158 Geometry of Banach spaces, P.F.X. MÜLLER & W. SCHACHERMAYER (eds)
- 159 Groups St Andrews 1989 volume 1, C.M. CAMPBELL & E.F. ROBERTSON (eds)
- 160 Groups St Andrews 1989 volume 2, C.M. CAMPBELL & E.F. ROBERTSON (eds)
- 161 Lectures on block theory, BURKHARD KÜLSHAMMER
- 163 Topics in varieties of group representations, S.M. VOVSI
- 164 Quasi-symmetric designs, M.S. SHRIKANDÉ & S.S. SANE
- 166 Surveys in combinatorics, 1991, A.D. KEEDWELL (ed)
- 168 Representations of algebras, H. TACHIKAWA & S. BRENNER (eds)
- 169 Boolean function complexity, M.S. PATERSON (ed)
- 170 Manifolds with singularities and the Adams-Novikov spectral sequence, B. BOTVINNIK
- 171 Squares, A.R. RAJWADE
- 172 Algebraic varieties, GEORGE R. KEMPF
- 173 Discrete groups and geometry, W.J. HARVEY & C. MACLACHLAN (eds)
- 174 Lectures on mechanics, J.E. MARSDEN
- 175 Adams memorial symposium on algebraic topology 1, N. RAY & G. WALKER (eds)
- 176 Adams memorial symposium on algebraic topology 2, N. RAY & G. WALKER (eds)
- 177 Applications of categories in computer science, M. FOURMAN, P. JOHNSTONE & A. PITTS (eds)
- 178 Lower K- and L-theory, A. RANICKI
- 179 Complex projective geometry, G. ELLINGSRUD *et al*
- 180 Lectures on ergodic theory and Pesin theory on compact manifolds, M. POLLICOTT
- 181 Geometric group theory I, G.A. NIBLO & M.A. ROLLER (eds)
- 182 Geometric group theory II, G.A. NIBLO & M.A. ROLLER (eds)
- 183 Shintani zeta functions, A. YUKIE
- 184 Arithmetical functions, W. SCHWARZ & J. SPILKER
- 185 Representations of solvable groups, O. MANZ & T.R. WOLF
- 186 Complexity: knots, colourings and counting, D.J.A. WELSH
- 187 Surveys in combinatorics, 1993, K. WALKER (ed)
- 188 Local analysis for the odd order theorem, H. BENDER & G. GLAUBERMAN
- 189 Locally presentable and accessible categories, J. ADAMEK & J. ROSICKY
- 190 Polynomial invariants of finite groups, D.J. BENSON
- 191 Finite geometry and combinatorics, F. DE CLERCK *et al*
- 192 Symplectic geometry, D. SALAMON (ed)
- 194 Independent random variables and rearrangement invariant spaces, M. BRAVERMAN
- 195 Arithmetic of blowup algebras, WOLMER VASCONCELOS
- 196 Microlocal analysis for differential operators, A. GRIGIS & J. SJÖSTRAND
- 197 Two-dimensional homotopy and combinatorial group theory, C. HOG-ANGELONI *et al*
- 198 The algebraic characterization of geometric 4-manifolds, J.A. HILLMAN
- 199 Invariant potential theory in the unit ball of C^d , MANFRED STOLL
- 200 The Grothendieck theory of dessins d'enfant, L. SCHNEPS (ed)
- 201 Singularities, JEAN-PAUL BRASSELET (ed)
- 202 The technique of pseudodifferential operators, H.O. CORDES
- 203 Hochschild cohomology of von Neumann algebras, A. SINCLAIR & R. SMITH
- 204 Combinatorial and geometric group theory, A.J. DUNCAN, N.D. GILBERT & J. HOWIE (eds)
- 205 Ergodic theory and its connections with harmonic analysis, K. PETERSEN & I. SALAMA (eds)
- 207 Groups of Lie type and their geometries, W.M. KANTOR & L. DI MARTINO (eds)
- 208 Vector bundles in algebraic geometry, N.J. HITCHIN, P. NEWSTEAD & W.M. OXBURY (eds)
- 209 Arithmetic of diagonal hypersurfaces over finite fields, F.Q. GOUVÉA & N. YUI
- 210 Hilbert C^* -modules, E.C. LANCE
- 211 Groups 93 Galway / St Andrews I, C.M. CAMPBELL *et al* (eds)
- 212 Groups 93 Galway / St Andrews II, C.M. CAMPBELL *et al* (eds)
- 214 Generalised Euler-Jacobi inversion formula and asymptotics beyond all orders, V. KOWALENKO *et al*
- 215 Number theory 1992–93, S. DAVID (ed)
- 216 Stochastic partial differential equations, A. ETHERIDGE (ed)
- 217 Quadratic forms with applications to algebraic geometry and topology, A. PFISTER
- 218 Surveys in combinatorics, 1995, PETER ROWLINSON (ed)
- 220 Algebraic set theory, A. JOYAL & I. MOERDIJK
- 221 Harmonic approximation, S.J. GARDINER
- 222 Advances in linear logic, J.-Y. GIRARD, Y. LAFONT & L. REGNIER (eds)
- 223 Analytic semigroups and semilinear initial boundary value problems, KAZUAKI TAIRA
- 224 Computability, enumerability, unsolvability, S.B. COOPER, T.A. SLAMAN & S.S. WAINER (eds)
- 225 A mathematical introduction to string theory, S. ALBEVERIO *et al*
- 226 Novikov conjectures, index theorems and rigidity I, S. FERRY, A. RANICKI & J. ROSENBERG (eds)
- 227 Novikov conjectures, index theorems and rigidity II, S. FERRY, A. RANICKI & J. ROSENBERG (eds)
- 228 Ergodic theory of Z^d actions, M. POLLICOTT & K. SCHMIDT (eds)
- 229 Ergodicity for infinite dimensional systems, G. DA PRATO & J. ZABCZYK
- 230 Prolegomena to a middlebrow arithmetic of curves of genus 2, J.W.S. CASSELS & E.V. FLYNN

- 231 Semigroup theory and its applications, K.H. HOFMANN & M.W. MISLOVE (eds)
- 232 The descriptive set theory of Polish group actions, H. BECKER & A.S. KECHRIS
- 233 Finite fields and applications, S. COHEN & H. NIEDERREITER (eds)
- 234 Introduction to subfactors, V. JONES & V.S. SUNDER
- 235 Number theory 1993–94, S. DAVID (ed)
- 236 The James forest, H. FETTER & B. GAMBOA DE BUEN
- 237 Sieve methods, exponential sums, and their applications in number theory, G.R.H. GREAVES *et al*
- 238 Representation theory and algebraic geometry, A. MARTSINKOVSKY & G. TODOROV (eds)
- 240 Stable groups, FRANK O. WAGNER
- 241 Surveys in combinatorics, 1997, R.A. BAILEY (ed)
- 242 Geometric Galois actions I, L. SCHNEPS & P. LOCHAK (eds)
- 243 Geometric Galois actions II, L. SCHNEPS & P. LOCHAK (eds)
- 244 Model theory of groups and automorphism groups, D. EVANS (ed)
- 245 Geometry, combinatorial designs and related structures, J.W.P. HIRSCHFELD *et al*
- 246 p -Automorphisms of finite p -groups, E.I. KHUKHRO
- 247 Analytic number theory, Y. MOTOHASHI (ed)
- 248 Tame topology and o -minimal structures, LOU VAN DEN DRIES
- 249 The atlas of finite groups: ten years on, ROBERT CURTIS & ROBERT WILSON (eds)
- 250 Characters and blocks of finite groups, G. NAVARRO
- 251 Gröbner bases and applications, B. BUCHBERGER & F. WINKLER (eds)
- 252 Geometry and cohomology in group theory, P. KROPHOLLER, G. NIBLO, R. STÖHR (eds)
- 253 The q -Schur algebra, S. DONKIN
- 254 Galois representations in arithmetic algebraic geometry, A.J. SCHOLL & R.L. TAYLOR (eds)
- 255 Symmetries and integrability of difference equations, P.A. CLARKSON & F.W. NIJHOFF (eds)
- 256 Aspects of Galois theory, HELMUT VÖLKLEIN *et al*
- 257 An introduction to noncommutative differential geometry and its physical applications 2ed, J. MADORE
- 258 Sets and proofs, S.B. COOPER & J. TRUSS (eds)
- 259 Models and computability, S.B. COOPER & J. TRUSS (eds)
- 260 Groups St Andrews 1997 in Bath, I, C.M. CAMPBELL *et al*
- 261 Groups St Andrews 1997 in Bath, II, C.M. CAMPBELL *et al*
- 262 Analysis and logic, C.W. HENSON, J. IOVINO, A.S. KECHRIS & E. ODELL
- 263 Singularity theory, BILL BRUCE & DAVID MOND (eds)
- 264 New trends in algebraic geometry, K. HULEK, F. CATANESE, C. PETERS & M. REID (eds)
- 265 Elliptic curves in cryptography, I. BLAKE, G. SEROUSSI & N. SMART
- 267 Surveys in combinatorics, 1999, J.D. LAMB & D.A. PREECE (eds)
- 268 Spectral asymptotics in the semi-classical limit, M. DIMASSI & J. SJÖSTRAND
- 269 Ergodic theory and topological dynamics, M.B. BEKKA & M. MAYER
- 270 Analysis on Lie groups, N.T. VAROPOULOS & S. MUSTAPHA
- 271 Singular perturbations of differential operators, S. ALBEVERIO & P. KURASOV
- 272 Character theory for the odd order theorem, T. PETERFALVI
- 273 Spectral theory and geometry, E.B. DAVIES & Y. SAFAROV (eds)
- 274 The Mandlebrot set, theme and variations, TAN LEI (ed)
- 275 Descriptive set theory and dynamical systems, M. FOREMAN *et al*
- 276 Singularities of plane curves, E. CASAS-ALVERO
- 277 Computational and geometric aspects of modern algebra, M.D. ATKINSON *et al*
- 278 Global attractors in abstract parabolic problems, J.W. CHOLEWA & T. DLOTKO
- 279 Topics in symbolic dynamics and applications, F. BLANCHARD, A. MAASS & A. NOGUEIRA (eds)
- 280 Characters and automorphism groups of compact Riemann surfaces, THOMAS BREUER
- 281 Explicit birational geometry of 3-folds, ALESSIO CORTI & MILES REID (eds)
- 282 Auslander-Buchweitz approximations of equivariant modules, M. HASHIMOTO
- 283 Nonlinear elasticity, Y. FU & R.W. OGDEN (eds)
- 284 Foundations of computational mathematics, R. DEVORE, A. ISERLES & E. SÜLI (eds)
- 285 Rational points on curves over finite fields, H. NIEDERREITER & C. XING
- 286 Clifford algebras and spinors 2ed, P. LOUNESTO
- 287 Topics on Riemann surfaces and Fuchsian groups, E. BUJALANCE, A.F. COSTA & E. MARTÍNEZ (eds)
- 288 Surveys in combinatorics, 2001, J. HIRSCHFELD (ed)
- 289 Aspects of Sobolev-type inequalities, L. SALOFF-COSTE
- 290 Quantum groups and Lie theory, A. PRESSLEY (ed)
- 291 Tits buildings and the model theory of groups, K. TENT (ed)
- 292 A quantum groups primer, S. MAJID
- 293 Second order partial differential equations in Hilbert spaces, G. DA PRATO & J. ZABCYK
- 294 Introduction to the theory of operator spaces, G. PISIER
- 295 Geometry and integrability, LIONEL MASON & YAVUZ NUTKU (eds)
- 296 Lectures on invariant theory, IGOR DOLGACHEV
- 297 The homotopy category of simply connected 4-manifolds, H.-J. BAUES
- 299 Kleinian groups and hyperbolic 3-manifolds, Y. KOMORI, V. MARKOVIC, & C. SERIES (eds)
- 300 Introduction to Möbius differential geometry, UDO HERTRICH-JEROMIN
- 301 Stable modules and the D(2)-problem, F.E.A. JOHNSON
- 302 Discrete and continuous nonlinear Schrödinger systems, M.J. ABLOWITZ, B. PRINARI, & A.D. TRUBATCH
- 303 Number theory and algebraic geometry, MILES REID & ALEXEI SKOROBOGATOV (eds)
- 304 Groups St Andrews 2001 in Oxford Vol. 1, COLIN CAMPBELL, EDMUND ROBERTSON & GEOFF SMITH (eds)
- 305 Groups St Andrews 2001 in Oxford Vol. 2, C.M. CAMPBELL, E.F. ROBERTSON & G.C. SMITH (eds)
- 307 Surveys in combinatorics 2003, C.D. WENSLEY (ed)
- 309 Corings and comodules, TOMASZ BRZEZINSKI & ROBERT WISBAUER
- 310 Topics in dynamics and ergodic theory, SERGEY BEZUGLYI & SERGIY KOLYADA (eds)
- 312 Foundations of computational mathematics, Minneapolis 2002, FELIPE CUCKER *et al* (eds)

London Mathematical Society Lecture Note Series. 317

Advances in Elliptic Curve Cryptography

Edited by

Ian F. Blake

University of Toronto

Gadiel Seroussi

Hewlett-Packard Laboratories

Nigel P. Smart

University of Bristol



CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS

Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press

The Edinburgh Building, Cambridge CB2 2RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org

Information on this title: www.cambridge.org/9780521604154

© Cambridge University Press 2005

This book is in copyright. Subject to statutory exception and to the provision of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published in print format 2005

ISBN-13 978-0-511-11161-7 eBook (MyiLibrary)

ISBN-10 0-511-11161-4 eBook (MyiLibrary)

ISBN-13 978-0-521-60415-4 paperback

ISBN-10 0-521-60415-X paperback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this book, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

Contents

Preface	<i>page</i> ix
Abbreviations and Standard Notation	xi
Authors	xv
Part 1. Protocols	
Chapter I. Elliptic Curve Based Protocols	
<i>N.P. Smart</i>	3
I.1. Introduction	3
I.2. ECDSA	4
I.3. ECDH/ECMQV	8
I.4. ECIES	12
I.5. Other Considerations	18
Chapter II. On the Provable Security of ECDSA	
<i>D. Brown</i>	21
II.1. Introduction	21
II.2. Definitions and Conditions	23
II.3. Provable Security Results	32
II.4. Proof Sketches	33
II.5. Further Discussion	36
Chapter III. Proofs of Security for ECIES	
<i>A.W. Dent</i>	41
III.1. Definitions and Preliminaries	42
III.2. Security Proofs for ECIES	50
III.3. Other Attacks Against ECIES	58
III.4. ECIES-KEM	61

Part 2. Implementation Techniques

Chapter IV. Side-Channel Analysis

E. Oswald

	69
IV.1. Cryptographic Hardware	70
IV.2. Active Attacks	71
IV.3. Passive Attacks	72
IV.4. Simple SCA Attacks on Point Multiplications	77
IV.5. Differential SCA Attacks on Point Multiplications	84

Chapter V. Defences Against Side-Channel Analysis

M. Joye

	87
V.1. Introduction	87
V.2. Indistinguishable Point Addition Formulæ	88
V.3. Regular Point Multiplication Algorithms	93
V.4. Base-Point Randomization Techniques	97
V.5. Multiplier Randomization Techniques	98
V.6. Preventing Side-Channel Analysis	100

Part 3. Mathematical Foundations

Chapter VI. Advances in Point Counting

F. Vercauteren

	103
VI.1. p -adic Fields and Extensions	104
VI.2. Satoh's Algorithm	105
VI.3. Arithmetic Geometric Mean	115
VI.4. Generalized Newton Iteration	121
VI.5. Norm Computation	128
VI.6. Concluding Remarks	132

Chapter VII. Hyperelliptic Curves and the HCDLP

P. Gaudry

	133
VII.1. Generalities on Hyperelliptic Curves	133
VII.2. Algorithms for Computing the Group Law	136
VII.3. Classical Algorithms for HCDLP	140
VII.4. Smooth Divisors	142
VII.5. Index-Calculus Algorithm for Hyperelliptic Curves	144
VII.6. Complexity Analysis	146
VII.7. Practical Considerations	149

Chapter VIII. Weil Descent Attacks

F. Hess

	151
VIII.1. Introduction – the Weil Descent Methodology	151
VIII.2. The GHS Attack	153
VIII.3. Extending the GHS Attack Using Isogenies	166

VIII.4. Summary of Practical Implications	173
VIII.5. Further Topics	175

Part 4. Pairing Based Techniques

Chapter IX. Pairings

S. Galbraith

	183
IX.1. Bilinear Pairings	183
IX.2. Divisors and Weil Reciprocity	184
IX.3. Definition of the Tate Pairing	185
IX.4. Properties of the Tate Pairing	187
IX.5. The Tate Pairing over Finite Fields	189
IX.6. The Weil Pairing	191
IX.7. Non-degeneracy, Self-pairings and Distortion Maps	192
IX.8. Computing the Tate Pairing Using Miller's Algorithm	196
IX.9. The MOV/Frey–Rück Attack on the ECDLP	197
IX.10. Supersingular Elliptic Curves	198
IX.11. Applications and Computational Problems from Pairings	201
IX.12. Parameter Sizes and Implementation Considerations	203
IX.13. Suitable Supersingular Elliptic Curves	204
IX.14. Efficient Computation of the Tate Pairing	205
IX.15. Using Ordinary Curves	208
Appendix: Proof of Weil Reciprocity	212

Chapter X. Cryptography from Pairings

K.G. Paterson

	215
X.1. Introduction	215
X.2. Key Distribution Schemes	218
X.3. Identity-Based Encryption	221
X.4. Signature Schemes	228
X.5. Hierarchical Identity-Based Cryptography and Related Topics	235
X.6. More Key Agreement Protocols	240
X.7. Applications and Infrastructures	242
X.8. Concluding Remarks	250

Bibliography

	253
Summary of Major LNCS Proceedings	271

Author Index

273

Subject Index

277

Preface

It is now more than five years since we started working on the book *Elliptic Curves in Cryptography* and more than four years since it was published. We therefore thought it was time to update the book since a lot has happened in the intervening years. However, it soon became apparent that a simple update would not be sufficient since so much has been developed in this area. We therefore decided to develop a second volume by inviting leading experts to discuss issues which have arisen.

Highlights in the intervening years which we cover in this volume include:

Provable Security. There has been considerable work in the last few years on proving various practical encryption and signature schemes secure. In this new volume we will examine the proofs for the ECDSA signature scheme and the ECIES encryption scheme.

Side-Channel Analysis. The use of power and timing analysis against cryptographic tokens, such as smart cards, is particularly relevant to elliptic curves since elliptic curves are meant to be particularly suited to the constrained environment of smart cards. We shall describe what side-channel analysis is and how one can use properties of elliptic curves to defend against it.

Point Counting. In 1999 the only method for computing the group order of an elliptic curve was the Schoof-Elkies-Atkin algorithm. However, for curves over fields of small characteristic we now have the far more efficient Satoh method, which in characteristic two can be further simplified into the AGM-based method of Mestre. We shall describe these improvements in this book.

Weil Descent. Following a talk by Frey in 1999, there has been considerable work on showing how Weil descent can be used to break certain elliptic curve systems defined over “composite fields” of characteristic two.

Pairing-Based Cryptography. The use of the Weil and Tate pairings was until recently confined to breaking elliptic curve protocols. But since the advent of Joux’s tripartite Diffie–Hellman protocol there has been an interest in using pairings on elliptic curves to construct protocols which cannot be implemented in another way. The most spectacular example of this is the

identity-based encryption algorithm of Boneh and Franklin. We describe not only these protocols but how these pairings can be efficiently implemented.

As one can see once again, the breadth of subjects we cover will be of interest to a wide audience, including mathematicians, computer scientists and engineers. Once again we also do not try to make the entire book relevant to all audiences at once but trust that, whatever your interests, you can find something of relevance within these pages.

The overall style and notation of the first book is retained, and we have tried to ensure that our experts have coordinated what they write to ensure a coherent account across chapters.

Ian Blake
Gadiel Seroussi
Nigel Smart

Abbreviations and Standard Notation

Abbreviations

The following abbreviations of standard phrases are used throughout the book:

AES	Advanced Encryption Standard
AGM	Arithmetic Geometric Mean
BDH	Bilinear Diffie–Hellman problem
BSGS	Baby Step/Giant Step method
CA	Certification Authority
CCA	Chosen Ciphertext Attack
CDH	Computational Diffie–Hellman problem
CM	Complex Multiplication
CPA	Chosen Plaintext Attack
DBDH	Decision Bilinear Diffie–Hellman problem
DDH	Decision Diffie–Hellman problem
DEM	Data Encapsulation Mechanism
DHAES	Diffie–Hellman Augmented Encryption Scheme
DHIES	Diffie–Hellman Integrated Encryption Scheme
DHP	Diffie–Hellman Problem
DLP	Discrete Logarithm Problem
DPA	Differential Power Analysis
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ECDDH	Elliptic Curve Decision Diffie–Hellman problem
ECDH	Elliptic Curve Diffie–Hellman protocol
ECDHP	Elliptic Curve Diffie–Hellman Problem
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
ECMQV	Elliptic Curve Menezes–Qu–Vanstone protocol
GHS	Gaudry–Hess–Smart attack
GRH	Generalized Riemann Hypothesis
HCDLP	Hyperelliptic Curve Discrete Logarithm Problem
HIBE	Hierarchical Identity-Based Encryption

IBE	Identity-Based Encryption
IBSE	Identity-Based Sign and Encryption
ILA	Information Leakage Analysis
KDF	Key Derivation Function
KDS	Key Distribution System
KEM	Key Encapsulation Mechanism
MAC	Message Authentication Code
MOV	Menezes–Okamoto–Vanstone attack
NIKDS	Non-Interactive Key Distribution System
PKI	Public Key Infrastructure
RSA	Rivest–Shamir–Adleman encryption scheme
SCA	Side Channel Analysis
SEA	Schoof–Elkies–Atkin algorithm
SHA	Secure Hash Algorithm
SPA	Simple Power Analysis
SSCA	Simple Side-Channel Attack
TA	Trusted Authority

Standard notation

The following standard notation is used throughout the book, often without further definition. Other notation is defined locally near its first use.

Basic Notation

$\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	integers, rationals, reals and complex numbers
$\mathbb{Z}_{>k}$	integers greater than k ; similarly for $\geq, <, \leq$
$\mathbb{Z}/n\mathbb{Z}$	integers modulo n
$\#S$	cardinality of the set S
$\gcd(f, g), \operatorname{lcm}(f, g)$	GCD, LCM of f and g
$\deg(f)$	degree of a polynomial f
ϕ_{Eul}	Euler totient function
$\left(\frac{\cdot}{p}\right)$	Legendre symbol
$\log_b x$	logarithm to base b of x ; natural log if b omitted
$O(f(n))$	function $g(n)$ such that $ g(n) \leq c f(n) $ for some constant $c > 0$ and all sufficiently large n
$o(f(n))$	function $g(n)$ such that $\lim_{n \rightarrow \infty} (g(n)/f(n)) = 0$
$\mathbb{P}^n$	projective space

Group/Field Theoretic Notation

$\mathbb{F}_q$	finite field with q elements
$K^*, K^+, \overline{K}$	for a field K , the multiplicative group, additive group and algebraic closure, respectively
$\operatorname{char}(K)$	characteristic of K
$\langle g \rangle$	cyclic group generated by g
$\operatorname{ord}(g)$	order of an element g in a group
$\operatorname{Aut}(G)$	automorphism group of G
$\mathbb{Z}_p, \mathbb{Q}_p$	p -adic integers and numbers, respectively
$\operatorname{Tr}_{q p}(x)$	trace of $x \in \mathbb{F}_q$ over $\mathbb{F}_p$, $q = p^n$
μ_n	n th roots of unity
$N_{L/K}$	norm map

Function Field Notation

$\deg(D)$	degree of a divisor
(f)	divisor of a function
$f(D)$	function evaluated at a divisor
$\sim$	equivalence of divisors
$\operatorname{ord}_P(f)$	multiplicity of a function at a point

Galois Theory Notation

$\operatorname{Gal}(K/F)$	Galois group of K over F
$\sigma(P)$	Galois conjugation of point P by σ
f^σ	Galois conjugation of coefficients of function f by σ

Curve Theoretic Notation

E	elliptic curve (equation)
(x_P, y_P)	coordinates of the point P
$x(P)$	the x -coordinate of the point P
$y(P)$	the y -coordinate of the point P
$E(K)$	group of K -rational points on E
$[m]P$	multiplication-by- m map applied to the point P
$E[m]$	group of m -torsion points on the elliptic curve E
$\text{End}(E)$	endomorphism ring of E
$\mathcal{O}$	point at infinity (on an elliptic curve)
$\wp$	Weierstraß ‘pay’ function
φ	Frobenius map
$\langle P, Q \rangle_n$	Tate pairing of P and Q
$e_n(P, Q)$	Weil pairing of P and Q
$e(P, Q)$	pairing of P and Q
$\hat{e}(P, Q)$	modified pairing of P and Q
$\text{Tr}(P)$	trace map
$\mathcal{T}$	trace zero subgroup

Authors

We would like to acknowledge the following people who contributed chapters to this book.

Dan Brown,
Certicom Corp.,
Mississauga,
Canada.

Alex Dent,
Mathematics Department,
Royal Holloway,
University of London,
United Kingdom.

Steven Galbraith,
Mathematics Department,
Royal Holloway,
University of London,
United Kingdom.

Pierrick Gaudry,
Laboratoire d'Informatique (LIX),
École Polytechnique ,
France.

Florian Hess,
Institut für Mathematik,
T.U. Berlin,
Germany.

Marc Joye,
Card Security Group,
Gemplus,
France.

Elisabeth Oswald,
Institute for Applied Information
Processing and Communications,
Graz University of Technology,
Austria.

Kenneth G. Paterson,
Info. Sec. Group,
Royal Holloway,
University of London,
United Kingdom.

Nigel Smart,
Department of Computer Sci-
ence,
University of Bristol,
United Kingdom.

Frederik Vercauteren,
Department of Computer Science,
University of Bristol,
United Kingdom.

The editors would like to thank Marc Joye for various bits of LaTeX help and Georgina Cranshaw and Ian Holyer for organizing our system for exchanging various files and keeping things up to date. As always, Roger Astley

of Cambridge University Press was very helpful throughout the whole process.

The authors of each chapter would like to thank the following for helping in checking and in the creation of their respective chapters:

- **Nigel Smart:** Alex Dent and Dan Brown.
- **Dan Brown:** Nigel Smart, Alex Dent, Kenneth Patterson and Ian Blake.
- **Alex Dent:** Bill and Jean Dent, Steven Galbraith, Becky George, Louis Granboulan, Victor Shoup, Andrew Spicer and Christine Swart (twice).
- **Steven Galbraith:** Paulo Barreto, Dan Boneh, Young-Ju Choie, Keith Harrison, Florian Hess, Neal Koblitz, Wenbo Mao, Kim Nguyen, Kenny Paterson, Maura Paterson, Hans-Georg Rück, Adam Saunders, Alice Silverberg, Lawrence Washington, Annegret Weng, Bill Williams and The Nuffield Foundation (Grant NUF-NAL 02).
- **Elisabeth Oswald:** The power traces presented in this chapter were made with the FPGA measurement-setup which was built by Siddika Berna Örs and has been presented in [268].
- **Marc Joye:** Benoît Chevallier-Mames and Tanja Lange.
- **Kenneth G. Paterson:** Sattam Al-Riyami, Alex Dent, Steven Galbraith, Caroline Kudla and The Nuffield Foundation (Grant NUF-NAL 02).

Part 1

Protocols

CHAPTER I

Elliptic Curve Based Protocols

N.P. Smart

I.1. Introduction

In this chapter we consider the various cryptographic protocols in which elliptic curves are primarily used. We present these in greater detail than in the book [ECC] and focus on their cryptographic properties. We shall only focus on three areas: signatures, encryption and key agreement. For each of these areas we present the most important protocols, as defined by various standard bodies.

The standardization of cryptographic protocols, and elliptic curve protocols in particular, has come a long way in the last few years. Standardization is important if one wishes to deploy systems on a large scale, since different users may have different hardware/software combinations. Working to a well-defined standard for any technology aids interoperability and so should aid the takeup of the technology.

In the context of elliptic curve cryptography, standards are defined so that one knows not only the precise workings of each algorithm, but also the the format of the transmitted data. For example, a standard answers such questions as

- In what format are finite field elements and elliptic curve points to be transmitted?
- How are public keys to be formatted before being signed in a certificate?
- How are conversions going to be performed between arbitrary bit strings to elements of finite fields, or from finite field elements to integers, and vice versa?
- How are options such as the use of point compression, (see [ECC, Chapter VI]) or the choice of curve to be signalled to the user?

A number of standardization efforts have taken place, and many of these reduce the choices available to an implementor by recommending or mandating certain parameters, such as specific curves and/or specific finite fields. This not only helps aid interoperability, it also means that there are well-defined sets of parameter choices that experts agree provide a given security level. In addition, by recommending curves it means that not every one who wishes to deploy elliptic curve based solutions needs to implement a point counting method like those in Chapter VI or [ECC, Chapter VII]. Indeed, since many

curves occur in more than one standard, if one selects a curve from the intersection then, your system will more likely interoperate with people who follow a different standard from you.

Of particular relevance to elliptic curve cryptography are the following standards:

- **IEEE 1363:** This standard contains virtually all public-key algorithms. In particular, it covers ECDH, ECDSA, ECMQV and ECIES, all of which we discuss in this chapter. In addition, this standard contains a nice appendix covering all the basic number-theoretic algorithms required for public-key cryptography.
- **ANSI X9.62 and X9.63:** These two standards focus on elliptic curves and deal with ECDSA in X9.62 and ECDH, ECMQV and ECIES in X9.63. They specify both the message formats to be used and give a list of recommended curves.
- **FIPS 186.2:** This NIST standard for digital signatures is an update of the earlier FIPS 186 [**FIPS 186**], which details the DSA algorithm only. FIPS 186.2 specifies both DSA and ECDSA and gives a list of recommended curves, which are mandated for use in U.S. government installations.
- **SECG:** The SECG standard was written by an industrial group led by Certicom. It essentially mirrors the contents of the ANSI standards but is more readily available on the Web, from the site
<http://www.secg.org/>
- **ISO:** There are two relevant ISO standards: ISO 15946-2, which covers ECDSA and a draft ISO standard covering a variant of ECIES called ECIES-KEM; see [305].

I.2. ECDSA

ECDSA is the elliptic curve variant of the Digital Signature Algorithm (DSA) or, as it is sometimes called, the Digital Signature Standard (DSS). Before presenting ECDSA it may be illustrative to describe the original DSA so one can see that it is just a simple generalization.

In DSA one first chooses a hash function H that outputs a bit-string of length m bits. Then one defines a prime q , of over m bits, and a prime p of n bits such that

- q divides $p - 1$.
- The discrete logarithm problem in the subgroup of $\mathbb{F}_p$ of order q is infeasible.

With current techniques and computing technology, this second point means that n should be at least 1024. Whilst to avoid birthday attacks on the hash function one chooses a value of m greater than 160.

One then needs to find a generator g for the subgroup of order q in $\mathbb{F}_p^*$. This is done by generating random elements $h \in \mathbb{F}_p^*$ and computing

$$g = h^{(p-1)/q} \pmod{p}$$

until one obtains a value of g that is not equal to 1. Actually, there is only a $1/q$ chance of this not working with the first h one chooses; hence finding a generator g is very simple.

Typically with DSA one uses SHA-1 [**FIPS 180.1**] as the hash function, although with the advent of SHA-256, SHA-384 and SHA-512 [**FIPS 180.2**] one now has a larger choice for larger values of m .

The quadruple (H, p, q, g) is called a set of domain parameters for the system, since they are often shared across a large number of users, e.g. a user domain. Essentially the domain parameters define a hash function, a group of order q , and a generator of this group.

The DSA makes use of the function

$$f : \begin{cases} \mathbb{F}_p^* & \longrightarrow & \mathbb{F}_q \\ x & \longmapsto & x \pmod{q}, \end{cases}$$

where one interprets $x \in \mathbb{F}_p^*$ as an integer when performing the reduction modulo q . This function is used to map group elements to integers modulo q and is often called the conversion function.

As a public/private-key pair in the DSA system one uses (y, x) where

$$y = g^x \pmod{p}.$$

The DSA signature algorithm then proceeds as follows:

ALGORITHM I.1: DSA Signing

INPUT: A message m and private key x .

OUTPUT: A signature (r, s) on the message m .

1. Choose $k \in_R \{1, \dots, q-1\}$.
 2. $t \leftarrow g^k \pmod{p}$.
 3. $r \leftarrow f(t)$.
 4. If $r = 0$ then goto Step 1.
 5. $e \leftarrow H(m)$
 6. $s \leftarrow (e + xr)/k \pmod{q}$
 7. If $s = 0$ then goto Step 1.
 8. Return (r, s) .
-

The verification algorithm is then given by

ALGORITHM I.2: DSA Verification

INPUT: A message m , a public key y and a signature (r, s) .

OUTPUT: Reject or Accept.

1. Reject if $r, s \notin \{1, \dots, q-1\}$.
 2. $e \leftarrow H(m)$.
 3. $u_1 \leftarrow e/s \pmod{q}$, $u_2 \leftarrow r/s \pmod{q}$.
 4. $t \leftarrow g^{u_1} y^{u_2} \pmod{p}$.
 5. Accept if and only if $r = f(t)$.
-

For ECDSA, the domain parameters are given by (H, K, E, q, G) , where H is a hash function, E is an elliptic curve over the finite field K , and G is a point on the curve of prime order q . Hence, the domain parameters again define a hash function, a group of order q , and a generator of this group. We shall always denote elliptic curve points by capital letters to aid understanding. With the domain parameters one also often stores the integer h , called the cofactor, such that

$$\#E(K) = h \cdot q.$$

This is because the value h will be important in other protocols and operations, which we shall discuss later. Usually one selects a curve such that $h \leq 4$.

The public/private-key pair is given by (Y, x) , where

$$Y = [x]G,$$

and the role of the function f is taken by

$$f : \begin{cases} E & \longrightarrow \\ P & \longmapsto x(P) \pmod{q}, \end{cases} \mathbb{F}_q$$

where $x(P)$ denotes the x -coordinate of the point P and we interpret this as an integer when performing the reduction modulo q . This interpretation is made even when the curve is defined over a field of characteristic two. In the case of even characteristic fields, one needs a convention as to how to convert an element in such a field, which is usually a binary polynomial $g(x)$, into an integer. Almost all standards adopt the convention that one simply evaluates $g(2)$ over the integers. Hence, the polynomial

$$x^5 + x^2 + 1$$

is interpreted as the integer 37, since

$$37 = 32 + 4 + 1 = 2^5 + 2^2 + 1.$$

The ECDSA algorithm then follows immediately from the DSA algorithm as:

ALGORITHM I.3: ECDSA Signing

INPUT: A message m and private key x .

OUTPUT: A signature (r, s) on the message m .

1. Choose $k \in_R \{1, \dots, q-1\}$.
 2. $T \leftarrow [k]G$.
 3. $r \leftarrow f(T)$.
 4. If $r = 0$ then goto Step 1.
 5. $e \leftarrow H(m)$
 6. $s \leftarrow (e + xr)/k \pmod{q}$.
 7. If $s = 0$ then goto Step 1.
 8. Return (r, s) .
-

The verification algorithm is then given by

ALGORITHM I.4: ECDSA Verification

INPUT: A message m , a public key Y and a signature (r, s) .

OUTPUT: Reject or Accept.

1. Reject if $r, s \notin \{1, \dots, q-1\}$.
 2. $e \leftarrow H(m)$.
 3. $u_1 \leftarrow e/s \pmod{q}$, $u_2 \leftarrow r/s \pmod{q}$.
 4. $T \leftarrow [u_1]G + [u_2]Y$.
 5. Accept if and only if $r = f(T)$.
-

One can show that ECDSA is provably secure, assuming that the elliptic curve group is modelled in a generic manner and H is a “good” hash function; see Chapter II for details.

An important aspect of both DSA and ECDSA is that the ephemeral secret k needs to be truly random. As a simple example of why this is so, consider the case where someone signs two different messages, m and m' , with the same value of k . The signatures are then (r, s) and (r', s') , where

$$\begin{aligned} r &= r' = f([k]G); \\ s &= (e + xr)/k \pmod{q}, \text{ where } e = H(m); \\ s' &= (e' + xr)/k \pmod{q}, \text{ where } e' = H(m'). \end{aligned}$$

We then have that

$$(e + xr)/s = k = (e' + xr)/s' \pmod{q}.$$

In which case we can deduce

$$xr(s' - s) = se' - s'e,$$

and hence

$$x = \frac{se' - s'e}{r(s' - s)} \pmod{q}.$$

So from now on we shall assume that each value of k is chosen at random.

In addition, due to a heuristic lattice attack of Howgrave-Graham and Smart [174], if a certain subset of the bits in k can be obtained by the attacker, then, over a number of signed messages, one can recover the long term secret x . This leakage of bits, often called *partial key exposure*, could occur for a number of reasons in practical systems, for example, by using a poor random number generator or by side-channel analysis (see Chapter IV for further details on side-channel analysis). The methods of Howgrave-Graham and Smart have been analysed further and extended by Nguyen and Shparlinski (see [261] and [262]). Another result along these lines is the attack of Bleichenbacher [31], who shows how a small bias in the random number generator, used to produce k , can lead to the recovery of the long-term secret x .

I.3. ECDH/ECMQV

Perhaps the easiest elliptic curve protocol to understand is the elliptic curve variant of the Diffie–Hellman protocol, ECDH. In this protocol two parties, usually called Alice and Bob, wish to agree on a shared secret over an insecure channel. They first need to agree on a set of domain parameters (K, E, q, h, G) as in our discussion on ECDSA. The protocol proceeds as follows:

Alice		Bob
a	$\xrightarrow{[a]G}$	$[a]G$
$[b]G$	$\xleftarrow{[b]G}$	b

Alice can now compute

$$K_A = [a]([b]G) = [ab]G$$

and Bob can now compute

$$K_B = [b]([a]G) = [ab]G.$$

Hence $K_A = K_B$ and both parties have agreed on the same secret key. The messages transferred are often referred to as *ephemeral public keys*, since they are of the form of discrete logarithm based public keys, but they exist for only a short period of time.

Given $[a]G$ and $[b]G$, the problem of recovering $[ab]G$ is called the Elliptic Curve Diffie–Hellman Problem, ECDHP. Clearly, if we can solve ECDLP then we can solve ECDHP; it is unknown if the other implication holds. A proof of equivalence of the DHP and DLP for many black box groups follows from the work of Boneh, Maurer and Wolf. This proof uses elliptic curves in a crucial way; see [ECC, Chapter IX] for more details.

The ECDH protocol has particularly small bandwidth if point compression is used and is very efficient compared to the standard, finite field based, Diffie–Hellman protocol.

The Diffie–Hellman protocol is a two-pass protocol, since there are two message flows in the protocol. The fact that both Alice and Bob need to be “online” to execute the protocol can be a problem in some situations. Hence, a one-pass variant exists in which only Alice sends a message to Bob. Bob’s ephemeral public key $[b]G$ now becomes a long-term static public key, and the protocol is simply a mechanism for Alice to transport a new session key over to Bob.

Problems can occur when one party does not send an element in the subgroup of order q . This can either happen by mistake or by design. To avoid this problem a variant called cofactor Diffie–Hellman is sometimes used. In cofactor Diffie–Hellman the shared secret is multiplied by the cofactor h before use, i.e., Alice and Bob compute

$$K_A = [h]([a]([b]G)) \text{ and } K_B = [h]([b]([a]G)).$$

The simplicity of the Diffie–Hellman protocol can however be a disguise, since in practice life is not so simple. For example, ECDH suffers from the man-in-the-middle attack:

Alice		Eve		Bob
a	$\xrightarrow{[a]G}$	$[a]G$		
$[x]G$	$\xleftarrow{[x]G}$	x		
		y	$\xrightarrow{[y]G}$	$[y]G$
		$[b]G$	$\xleftarrow{[b]G}$	b

In this attack, Alice agrees a key $K_A = [a]([x]G)$ with Eve, thinking it is agreed with Bob, and Bob agrees a key $K_B = [b]([y]G)$ with Eve, thinking it is agreed with Alice. Eve can now examine communications as they pass through her by essentially acting as a router.

The problem is that when performing ECDH we obtain no data-origin authentication. In other words, Alice does not know who the ephemeral public key she receives is from. One way to obtain data-origin authentication is to sign the messages in the Diffie–Hellman key exchange. Hence, for example, Alice must send to Bob the value

$$([a]G, (r, s)),$$

where (r, s) is her ECDSA signature on the message $[a]G$.

One should compare this model of authenticated key exchange with the traditional form of RSA-based key transport, as used in SSL. In RSA-based key transport, the RSA public key is used to encrypt a session key from one

user to the other. The use of a signed Diffie–Hellman key exchange has a number of advantages over an RSA-based key transport:

- In key transport only one party generates the session key, while in key agreement both can parties contribute randomness to the resulting session key.
- Signed ECDH has the property of forward secrecy, whereas an RSA-based key transport does not. An authenticated key agreement/transport protocol is called forward secure if the compromise of the long-term static key does not result in past session keys being compromised. RSA key transport is not forward secure since once you have the long-term RSA decryption key of the recipient you can determine the past session keys; however, in signed ECDH the long-term private keys are only used to produce signatures.

However, note that the one-pass variant of ECDH discussed above, being a key transport mechanism, also suffers from the above two problems of RSA key transport.

The problem with signed ECDH is that it is wasteful of bandwidth. To determine the session key we need to append a signature to the message flows. An alternative system is to return to the message flows in the original ECDH protocol but change the way that the session key is derived. If the session key is derived using static public keys, as well as the transmitted ephemeral keys, we can obtain *implicit* authentication of the resulting session key. This is the approach taken in the MQV protocol of Law, Menezes, Qu, Solinas and Vanstone [216].

In the MQV protocol both parties are assumed to have long-term static public/private key pairs. For example, we shall assume that Alice has the static key pair $([a]G, a)$ and Bob has the static key pair $([c]G, c)$. To agree on a shared secret, Alice and Bob generate two ephemeral key pairs; for example, Alice generates the ephemeral key pair $([b]G, b)$ and Bob generates the ephemeral key pair $([d]G, d)$. They exchange the public parts of these ephemeral keys as in the standard ECDH protocol:

$$\begin{array}{ccc}
 \text{Alice} & & \text{Bob} \\
 b & \xrightarrow{[b]G} & [b]G \\
 [d]G & \xleftarrow{[d]G} & d.
 \end{array}$$

Hence, the message flows are precisely the same as in the ECDH protocol. After the exchange of messages Alice knows

$$a, b, [a]G, [b]G, [c]G \text{ and } [d]G,$$

and Bob knows

$$c, d, [c]G, [d]G, [a]G \text{ and } [b]G.$$

The shared secret is then determined by Alice via the following algorithm:

ALGORITHM I.5: ECMQV Key Derivation

INPUT: A set of domain parameters (K, E, q, h, G)
 and $a, b, [a]G, [b]G, [c]G$ and $[d]G$.
 OUTPUT: A shared secret G ,
 shared with the entity with public key $[c]G$.

1. $n \leftarrow \lceil \log_2(\#K) \rceil / 2$.
2. $u \leftarrow (x([b]G) \pmod{2^n}) + 2^n$.
3. $s \leftarrow b + ua \pmod{q}$.
4. $v \leftarrow (x([d]G) \pmod{2^n}) + 2^n$.
5. $Q \leftarrow [s]([d]G + [v]([c]G))$.
6. If Q is at infinity goto Step 1.
7. Output Q .

Bob can also compute the same value of Q by swapping the occurrence of (a, b, c, d) in the above algorithm with (c, d, a, b) . If we let u_A, v_A and s_A denote the values of u, v and s computed by Alice and u_B, v_B and s_B denote the corresponding values computed by Bob, then we see

$$\begin{aligned} u_A &= v_B, \\ v_A &= u_B. \end{aligned}$$

We then see that

$$\begin{aligned} Q &= [s_A]([d]G + [v_A]([c]G)) \\ &= [s_A][d + v_A c]G \\ &= [s_A][d + u_B c]G \\ &= [s_A][s_B]G. \end{aligned}$$

In addition, a cofactor variant can be used by setting $Q \leftarrow [h]Q$ before the test for whether Q is the point at infinity in Step 6.

In summary, the ECMQV protocol allows authentic key agreement to occur over an insecure channel, whilst only requiring the same bandwidth as an unauthenticated Diffie–Hellman.

One can also have a one-pass variant of the ECMQV protocol, which enables one party to be offline when the key is agreed. Suppose Bob is the party who is offline; he will still have a long-term static public/private key pair given by $[c]G$. Alice then uses this public key both as the long-term key and the ephemeral key in the above protocol. Hence, Alice determines the shared secret via

$$Q = [s_A]([c]G + [v_A]([c]G)) = [s_A][v_A + 1]([c]G),$$

where, as before, $s_A = b + u_A a$, with a the long-term private key and b the ephemeral private key. Bob then determines the shared secret via

$$Q = [s_B] (([b]G) + v_B([a]G)),$$

where s_B is now fixed and equal to $(1 + u_B)c$.

It is often the case that a key agreement protocol also requires key confirmation. This means that both communicating parties know that the other party has managed to compute the shared secret. For ECMQV this is added by slightly modifying the protocol. Each party, on computing the shared secret point Q , then computes

$$(k, k') \leftarrow H(Q),$$

where H is a hash function (or key derivation function). The key k is used as the shared session key, whilst k' is used as a key to a Message Authentication Code, MAC, to enable key confirmation.

This entire procedure is accomplished in three passes as follows:

Alice	Bob
b	$[b]G$
$[d]G$	d

$$\begin{array}{c}
 \xrightarrow{[b]G} \\
 \xleftarrow{[d]G, M} \\
 \xrightarrow{M'}
 \end{array}$$

where

$$\begin{aligned}
 M &= \text{MAC}_{k'}(2, \text{Bob}, \text{Alice}, [d]G, [b]G), \\
 M' &= \text{MAC}_{k'}(3, \text{Alice}, \text{Bob}, [b]G, [d]G).
 \end{aligned}$$

Of course Alice needs to verify that M is correct upon receiving it, and Bob needs to do likewise for M' .

I.4. ECIES

The elliptic curve integrated encryption system (ECIES) is the standard elliptic curve based encryption algorithm. It is called integrated, since it is a hybrid scheme that uses a public-key system to transport a session key for use by a symmetric cipher. It is based on the DHAES/DHIES protocol of Abdalla, Bellare and Rogaway [1]. Originally called DHAES, for Diffie–Hellman Augmented Encryption Scheme, the name was changed to DHIES, for Diffie–Hellman Integrated Encryption Scheme, so as to avoid confusion with the AES, Advanced Encryption Standard.

ECIES is a public-key encryption algorithm. Like ECDSA, there is assumed to be a set of domain parameters (K, E, q, h, G) , but to these we also add a choice of symmetric encryption/decryption functions, which we shall denote $E_k(m)$ and $D_k(c)$. The use of a symmetric encryption function makes

it easy to encrypt long messages. In addition, instead of a simple hash function, we require two special types of hash functions:

- A message authentication code $MAC_k(c)$,

$$MAC : \{0, 1\}^n \times \{0, 1\}^* \longrightarrow \{0, 1\}^m.$$

This acts precisely like a standard hash function except that it has a secret key passed to it as well as a message to be hashed.

- A key derivation function $KD(T, l)$,

$$KD : E \times \mathbb{N} \longrightarrow \{0, 1\}^*.$$

A key derivation function acts precisely like a hash function except that the output length (the second parameter) could be quite large. The output is used as a key to encrypt a message; hence, if the key is to be used in a **xor**-based encryption algorithm the output needs to be as long as the message being encrypted.

The ECIES scheme works like a one-pass Diffie–Hellman key transport, where one of the parties is using a fixed long-term key rather than an ephemeral one. This is followed by symmetric encryption of the actual message. In the following we assume that the combined length of the required MAC key and the required key for the symmetric encryption function is given by l .

The recipient is assumed to have a long-term public/private-key pair (Y, x) , where

$$Y = [x]G.$$

The encryption algorithm proceeds as follows:

ALGORITHM I.6: ECIES Encryption

INPUT: Message m and public key Y .

OUTPUT: The ciphertext (U, c, r) .

1. Choose $k \in_R \{1, \dots, q-1\}$.
 2. $U \leftarrow [k]G$.
 3. $T \leftarrow [k]Y$.
 4. $(k_1 || k_2) \leftarrow KD(T, l)$.
 5. Encrypt the message, $c \leftarrow E_{k_1}(m)$.
 6. Compute the MAC on the ciphertext, $r \leftarrow MAC_{k_2}(c)$.
 7. Output (U, c, r) .
-

Each element of the ciphertext (U, c, r) is important:

- U is needed to agree the ephemeral Diffie–Hellman key T .
- c is the actual encryption of the message.
- r is used to avoid adaptive chosen ciphertext attacks.

Notice that the data item U can be compressed to reduce bandwidth, since it is an elliptic curve point.

Decryption proceeds as follows:

ALGORITHM I.7: ECIES Decryption

INPUT: Ciphertext (U, c, r) and a private key x .

OUTPUT: The message m or an “Invalid Ciphertext” message.

1. $T \leftarrow [x]U$.
 2. $(k_1 \| k_2) \leftarrow KD(T, l)$.
 3. Decrypt the message $m \leftarrow D_{k_1}(c)$.
 4. If $r \neq MAC_{k_2}(c)$ then output “Invalid Ciphertext”.
 5. Output m .
-

Notice that the T computed in the decryption algorithm is the same as the T computed in the encryption algorithm since

$$T_{\text{decryption}} = [x]U = [x]([k]G) = [k]([x]G) = [k]Y = T_{\text{encryption}}.$$

One can show that, assuming various properties of the block cipher, key derivation function and keyed hash function, the ECIES scheme is secure against adaptive chosen ciphertext attack, assuming a variant of the Diffie–Hellman problem in the elliptic curve group is hard; see [1] and Chapter III.

In many standards, the function KD is applied to the x -coordinate of the point T and not the point T itself. This is more efficient in some cases but leads to the scheme suffering from a problem called benign malleability. Benign malleability means that an adversary is able, given a ciphertext $\mathcal{C}$, to produce a different valid ciphertext $\mathcal{C}'$ of the same message. For ECIES, if $\mathcal{C} = (U, c, r)$, then $\mathcal{C}' = (-U, c, r)$ since if KD is only applied to the x -coordinate of U , so both $\mathcal{C}$ and $\mathcal{C}'$ are different valid ciphertexts corresponding to the same message.

The problem with benign malleability is that it means the scheme cannot be made secure under the formal definition of an adaptive chosen ciphertext attack. However, the issue is not that severe and can be solved, theoretically, by using a different but equally sensible definition of security. No one knows how to use the property of benign malleability in a “real-world” attack, and so whether one chooses a standard where KD is applied to T or just $x(T)$ is really a matter of choice.

In addition, to avoid problems with small subgroups, just as in the ECDH and ECMQV protocols, one can select to apply KD to either T or $[h]T$. The use of $[h]T$ means that the key derivation function is applied to an element in the group of order q , and hence if T is a point in the small subgroup one would obtain $[h]T = \mathcal{O}$.

The fact that ECIES suffers from benign malleability, and the fact that the cofactor variant can lead to interoperability problems, has led to a new approach being taken to ECIES in the draft ISO standard [305].

The more modern approach is to divide a public-key encryption algorithm into a key transport mechanism, called a Key Encapsulation Mechanism, or KEM, and a Data Encapsulation Mechanism, or DEM. This combined KEM/DEM approach has proved to be very popular in recent work because it divides the public key algorithm into two well-defined stages, which aids in the security analysis.

We first examine a generic DEM, which requires a MAC function MAC_k of key length n bits and a symmetric cipher E_k of key length m bits. The Data Encapsulation Mechanism then works as follows:

ALGORITHM I.8: DEM Encryption

INPUT: A key K of length $n + m$ bits and a message M .

OUTPUT: A ciphertext C

1. Parse K as $k_1 || k_2$,
where k_1 has m bits and k_2 has n bits.
 2. $c \leftarrow E_{k_1}(M)$.
 3. $r \leftarrow MAC_{k_2}(c)$.
 4. $C \leftarrow (c || r)$.
-

Decryption then proceeds as follows:

ALGORITHM I.9: DEM Decryption

INPUT: A key K of length $n + m$ bits and a ciphertext C .

OUTPUT: A message M or “Invalid Ciphertext”.

1. Parse K as $k_1 || k_2$,
where k_1 has m bits and k_2 has n bits.
 2. Parse C as $c || r$,
this could result in an “Invalid Ciphertext” warning.
 3. Decrypt the message $M \leftarrow D_{k_1}(c)$.
 4. If $r \neq MAC_{k_2}(c)$ then output “Invalid Ciphertext”.
 5. Output M .
-

To use a DEM we require a KEM, and we shall focus on one based on ECIES called ECIES-KEM. A KEM encryption function takes as input a public key and outputs a session key and the encryption of the session key under the given public key. The KEM decryption operation takes as input a private key and the output from a KEM encryption and produces the associated session key.

As mentioned before, the definition of ECIES-KEM in the draft ISO standard is slightly different from earlier versions of ECIES. In particular, the way the ephemeral secret is processed to deal with small subgroup attacks and how chosen ciphertext attacks are avoided is changed in the following scheme. The processing with the cofactor is now performed solely in the decryption phase, as we shall describe later. First we present the encryption phase for ECIES-KEM.

Again, the recipient is assumed to have a long-term public/private-key pair (Y, x) , where

$$Y = [x]G.$$

The encryption algorithm proceeds as follows:

ALGORITHM I.10: ECIES-KEM Encryption

INPUT: A public key Y and a length l .
 OUTPUT: A session key K of length l and
 an encryption E of K under Y .

1. Choose $k \in_R \{1, \dots, q-1\}$.
2. $E \leftarrow [k]G$.
3. $T \leftarrow [k]Y$.
4. $K \leftarrow KD(E||T, l)$,
5. Output (E, K) .

Notice how the key derivation function is applied to both the ephemeral public key and the point representing the session key. It is this modification that removes problems associated with benign malleability in chosen ciphertext attacks and aids in the security proof. In addition, no modification to the KEM is made when one wishes to deal with cofactors; this modification is only made at decryption time.

To deal with cofactors, suppose we have a set of domain parameters (K, E, q, h, G) . We set a flag f as follows:

- If $h = 1$, then $f \leftarrow 0$.
- If $h \neq 1$, then select $f \leftarrow 1$ or $f \leftarrow 2$.

We can now describe the ECIES-KEM decryption operation.

ALGORITHM I.11: ECIES-KEM Decryption

INPUT: An encryption session key E , a private key x ,
 a length l and a choice for the flag f as above.
 OUTPUT: A session key K of length l

1. If $f = 2$ then check whether E has order q ,
if not return ‘‘Invalid Ciphertext’’.
 2. $x' \leftarrow x$ and $E' \leftarrow E$.
 3. If $f = 1$ then
 4. $x' \leftarrow x'/h \pmod{q}$.
 5. $E' \leftarrow [h]E'$.
 6. $T \leftarrow [x']E'$.
 7. If $T = 0$ then return ‘‘Invalid Ciphertext’’.
 8. $K \leftarrow KD(E\|T, l)$,
 9. Output K .
-

We now explain how an encryption is performed with a KEM/DEM approach, where we are really focusing on using ECIES-KEM. We assume a KEM and DEM that are compatible, i.e., a pair whose KEM outputs an l -bit key and whose DEM requires an l -bit key as input.

ALGORITHM I.12: ECIES-KEM-DEM Encryption

INPUT: A public key Y , a message M .

OUTPUT: A ciphertext C .

1. $(E, K) \leftarrow ECIES - KEM_{Enc}(Y, l)$.
 2. $(c\|r) \leftarrow DEM_{Enc}(K, M)$.
 3. Output $(E\|c\|r)$.
-

ALGORITHM I.13: ECIES-KEM/DEM Decryption

INPUT: A ciphertext C , a private key x .

OUTPUT: A message m or ‘‘Invalid Ciphertext’’.

1. Parse C as $(E\|c\|r)$.
 2. $K \leftarrow ECIES - KEM_{Dec}(E, x, l)$.
 3. If K equals ‘‘Invalid Ciphertext’’ then
 4. Return ‘‘Invalid Ciphertext’’.
 5. $M \leftarrow DEM_{Dec}(K, (c\|r))$.
 6. If M equals ‘‘Invalid Ciphertext’’ then
 7. Return ‘‘Invalid Ciphertext’’.
 8. Output M .
-

I.5. Other Considerations

When receiving a public key, whether in a digital certificate or as an ephemeral key in ECDH, ECMQV or ECIES, one needs to be certain that the ephemeral key is a genuine point of the correct order on the given curve. This is often overlooked in many academic treatments of the subject.

The ANSI and SECG standards specify the following check, which should be performed for each received public key.

ALGORITHM I.14: Public-Key Validation

INPUT: A set of domain parameters (K, E, q, h, G)
 and a public key Q
 OUTPUT: Valid or Invalid

1. If $Q \notin E(K)$ then output ‘‘Invalid’’.
2. If $Q = \mathcal{O}$ then output ‘‘Invalid’’.
3. (Optional) If $[q]Q \neq \mathcal{O}$ then output ‘‘Invalid’’.
4. Output ‘‘Valid’’.

The last check is optional, because it can be quite expensive, especially if $h \neq 1$ in the case of large prime characteristic or $h \neq 2$ in the case of even characteristic. However, the check is needed to avoid problems with small subgroups. It is because this check can be hard to implement that the option of using cofactors in the key derivation functions is used in ECDH, ECMQV and ECIES.

Just as public keys need to be validated, there is also the problem of checking whether a given curve is suitable for use. The following checks should be performed before a set of domain parameters is accepted; however, this is likely to be carried out only once for each organization deploying elliptic curve based solutions.

ALGORITHM I.15: Elliptic Curve Validation

INPUT: A set of domain parameters (K, E, q, h, G)
 OUTPUT: Valid or Invalid

1. Let $l \leftarrow \#K = p^n$.
2. Check $\#E(K) = h \cdot q$, by generating random points and verifying that they have order h , q , or $h \cdot q$.
3. Check that q is prime.
4. Check that $q > 2^{160}$ to avoid the BSGS/Rho attacks, see [ECC, Chapter V] for details.
5. Check that $q \neq p$ to avoid the anomalous attack, again see [ECC, Chapter V] for reasons.
6. Check that $l^t \neq 1 \pmod{q}$ for all $t \leq 20$ to avoid the

MOV/Frey--Rück attack, see [ECC, Chapter V].

7. Check that n is prime, to avoid attacks based on Weil descent, see Chapter VIII of this volume.
8. Check that G lies on the curve and has order q .

But how do you know the curve has no special weakness known only to a small (clever) subset of people? Since we believe that such a weak curve must come from a very special small subset of all the possible curves, we generate the curve at random. But even if you generate your curve at random, you need to convince someone else that this is the case. This is done by generating the curve in a verifiably random way, which we shall now explain in the case of characteristic two curves. For other characteristics a similar method applies.

ALGORITHM I.16: Verifiable Random Generation of Curves

INPUT: A field $K = \mathbb{F}_{2^n}$ of characteristic two

OUTPUT: A set of domain parameters (K, E, q, h, G) and a seed S

1. Choose a random seed S .
2. Chain SHA-1 with input S to produce a bit string B of length n .
3. Let b be the element of K with bit representation B .
4. Set $E: Y^2 + X \cdot Y = X^3 + X^2 + b$.
5. Apply the methods of Chapter VI of this volume or [ECC, Chapter VII] to compute the group order $N \leftarrow \#E(K)$.
6. If $N \neq 2q$ with q prime then goto the Step 1.
7. Generate an element $G \in E(K)$ of order q .
8. Check that $(E, K, q, 2, G)$ passes Algorithm I.15, if not then goto Step 1.
9. Output $(K, E, q, 2, G)$ and S .

With the value of S , any other person can verify that the given elliptic curve is determined by S . Now if the generator knew of a subset of curves with a given weakness, to generate the appropriate S for a member of such a subset, they would need to be able to invert SHA-1, which is considered impossible.

CHAPTER II

On the Provable Security of ECDSA

D. Brown

II.1. Introduction

II.1.1. Background. The Elliptic Curve Digital Signature Algorithm is now in many standards or recommendations, such as [ANSI X9.62], [SECG], [FIPS 186.2], [IEEE 1363], [ISO 15946-2], [NESSIE] and [RFC 3278]. Organizations chose ECDSA because they regarded its reputational security sufficient, on the grounds that (a) it is a very natural elliptic curve analogue of DSA, and that (b) both elliptic curve cryptography and DSA were deemed to have sufficiently high reputational security. The standardization of ECDSA has created more intense public scrutiny. Despite this, no substantial weaknesses in ECDSA have been found, and thus its reputational security has increased.

At one point, proofs of security, under certain assumptions, were found for digital signature schemes similar to DSA and ECDSA. The proof techniques in these initial proofs did not, and still do not, appear applicable to DSA and ECDSA. Thus, for a time, provable security experts suggested a change to the standardization of reputationally secure schemes, because slight modifications could improve provable security.

Further investigation, however, led to new provable security results for ECDSA. New proof techniques and assumptions were found that overcame or avoided the difficulty in applying the initial techniques to ECDSA. This chapter describes some of these results, sketches their proofs, and discusses the impact and interpretation of these results.

Interestingly, in some cases, the new proof techniques did not apply to DSA, which was the first, though mild, indication that ECDSA may have better security than DSA. Furthermore, some of the new proof techniques do not work for the modified versions of ECDSA for which the initial proof techniques applied. Therefore, it can no longer be argued that the modified versions have superior provable security; rather, it should be said that they have provable security incomparable to ECDSA.

Cryptanalysis results are the converse to provable security results and are just as important. In this chapter, conditional results are included, because no successful, practical cryptanalysis of ECDSA is known. The hypotheses of

a provable security result is a sufficient condition for security, while a cryptanalysis result establishes a necessary condition for security. For example, one conditional cryptanalysis result for ECDSA is that if a hash collision can be found, then a certain type of forgery of ECDSA is possible. Therefore, collision resistance of the message digest hash function is a necessary condition for the security of ECDSA. Note however that this is not yet a successful cryptanalysis of ECDSA, because no collisions have been found in ECDSA's hash function.

II.1.2. Examining the ECDSA Construction. The primary purpose of the provable security results are to examine the security of ECDSA. The purpose is not to examine the security of the primitives ECDSA uses (elliptic curve groups and hash functions). Even with the secure primitives, it does not follow a priori that a digital signature built from these primitives will be secure. Consider the following four signature scheme designs, characterized by their verification equations for signatures (r, s) . Each is based on ECDSA but with the value r used in various different ways, and in all cases signatures can be generated by the signer by computing $r = [k]G$ and applying a signing equation.

- The first scheme, with verification $r = f([s^{-1}r]([H(m)]G + Y))$, is forgeable through $(r, s) = (f([t]([H(m)]G + Y)), t^{-1}r)$, for any t and message m . Evidently, the verification equation does not securely bind, informally speaking, the five values r, s, m, G, Y .
- The second scheme, ECDSA, is verified with $r = f([s^{-1}]([H(m)]G + [r]Y))$. Moving the position of r on the right-hand side of the verification equation seems to turn an insecure scheme into a secure one. Now all five values have become securely bound.
- The third scheme, verified with $r = f([s^{-1}]([H(m, r)]G + Y))$, has r in yet another position. The third scheme seems secure, and the provable security results of Pointcheval and Stern [276] using the Forking Lemma seem adaptable to this scheme.
- A fourth scheme, verified with $r = f([s^{-1}]([H(m, r)]G + [r]Y))$, combines the second and third in that r appears twice on the right, once in each location of the second and third. Although the fourth scheme could well have better security than both the second and third schemes, it is argued that the overuse of r in the right-hand side of the third and fourth schemes is an obstacle to certain security proof techniques. Because the value r occurs both inside a hash function evaluation and as a scalar multiple in the elliptic curve group, formulating mild and independent hypotheses about the two primitives is not obvious and inhibits the construction of a security proof.

II.2. Definitions and Conditions

II.2.1. Security for Signatures. Goldwasser, Micali and Rivest introduced in [150] the now widely accepted formal definition for signature schemes and their security.

DEFINITION II.1 (Signature Scheme). *A signature scheme is a triple of probabilistic algorithms $\Sigma = (K, G, V)$, such that K has no input (except randomness) and outputs a public key Y and private key x ; G has input of the private key x and an arbitrary message m and outputs a signature S ; and V has input of the public key Y , message m and signature S and outputs either valid or invalid.*

A signature scheme is *correct* if the following holds: For any message m and any randomness, computing $K \mapsto (x, Y)$ and then $G : (x, m) \mapsto S$ will ensure the result $V : (Y, m, S) \mapsto \text{Valid}$. If G does not use its randomness input, then Σ is said to be deterministic. If, for each message m and public key Y , at most one signature S satisfies $V(Y, m, S) = \text{Valid}$, then Σ is said to be verifiably deterministic.

DEFINITION II.2. *A forger of signature scheme (K, G, V) is a probabilistic algorithm F , having input of either a public key Y or a signature S and an internal state X , and having output of a message m , state X , and R , which is either a signature or a request for a signature of a message m_i .*

A forger F is measured by its ability to win the following game.

DEFINITION II.3. *The forgery game for a forger F of signature scheme $\Sigma = (K, G, V)$ has multiple rounds, each consisting of two plays, the first by the signer and the second by the forger.*

- In Round 0, the signer uses K to output public key Y and a private key x .
- Next, the forger is given input of the public key Y and a fixed initial state X_0 , and it outputs a message m_i , a state X_1 and a request or signature R_1 .
- For $i \geq 1$, Round i works as follows.
 - If R_i is a request for a signature, then the signer uses G with input of x and the message m_i to output a signature S_i . Next, the forger is called again with input of the signature S_i and the state X_i . It will then return a new message m_{i+1} , a new state X_{i+1} , and a new request or signature R_{i+1} .
 - If R_i is a signature, not a request, then the game is over.

When the game ends, at Round i , the forger has won if both $m_{i+1} \neq m_1, \dots, m_i$ and $V(Y, m_{i+1}, R_{i+1}) = \text{Valid}$; otherwise, the forger has lost.

We can now define a meaningful forger.

DEFINITION II.4 (Forger). *A forger F is a (p, Q, t) -forger of signature scheme (K, G, V) if its probability of winning the forgery game in at most Q rounds using computational effort at most t is at least p . A signature Σ is (p, Q, t) -secure if it does not have a (p, Q, t) -forger.*

A $(p, 0, t)$ -forger is called a *passive* forger, and a forger that is not passive is called *active* when needed. It is important to realize that the forgers thus defined are successful without regard to the quality or meaningfulness of the forged message. To emphasize this limiting aspect, such forgers are called *existential* forgers. A *selective* forger, by contrast, lacks this limitation and is formally defined as follows.

DEFINITION II.5 (Selective Forger). *Let U be a probabilistic algorithm, with no input except randomness, and output of a message. A selective forger is a forger F with the following differences. The input of a public key also includes a message. The selective forgery game for a selective forger F of signature scheme (K, G, V) , with message selection oracle U , is the forgery game with the following differences. In Round 0, U is called to generate a message m_0 , which is given as input to F . The forger wins the game in Round i , only if $m_0 = m_{i+1}$ is satisfied. A selective forger F is a (p, Q, t, U) -forger of signature scheme (K, G, V) .*

A selective forger can forge any message from a certain class of messages. A selective forger is generally probably much more harmful than an existential forger, depending on the distribution of messages given by U . In particular, if U generates meaningful messages, then the selective forger can forge any meaningful message it wants.

Generally, with p and t the same, a passive forger is more harmful than an active one, and a selective forger is more harmful than an existential one. Generally, passiveness and selectiveness are qualitative attributes of forgers, and their importance depends on the usage of the signatures.

DEFINITION II.6 (Signature Security). *A signature scheme is (p, Q, t) -secure against existential forgery if there exists no (p, Q, t) -forger.*

A signature scheme is (p, Q, t, U) -secure against selective forgery if there exists no (p, Q, t, U) -selective-forger.

II.2.2. Necessary Conditions. The conditions in Table II.1 on the components of ECDSA can be shown to be necessary for the security of ECDSA because otherwise forgery would be possible. These conditions are defined below together with some of the reductions to attacks that prove their necessity.

Intractable Semi-Logarithm : For a conversion function f and group $\langle G \rangle$, a semi-logarithm of a group element P to the base G is a pair of integers

TABLE II.1. Necessary Conditions with Associated Forgeries

Component	Condition	Forgery	
Group (and Conversion Function)	Intractable Discrete Logarithm	Passive	Selective
	Intractable Semi-Logarithm	Passive	Selective
Conversion Function	Almost Bijective	Passive	Selective
Random Number Generator	Arithmetically Unbiased	Active	Selective
Range Checking	Check $r \neq 0$	Passive	Selective
Hash Function	Rarely Zero	Passive	Selective
	Zero Resistant	Passive	Existential
	1st-Preimage Resistant	Passive	Existential
	2nd-Preimage Resistant	Active	Selective
	Collision Resistant	Active	Existential

(t, u) such that

$$t = f([u^{-1}](G + [t]P)).$$

Finding semi-logarithms needs to be intractable or else forgeries can be found by setting $P = [H(m)^{-1}]Y$, where Y is the public key, for then $(t, H(m)u)$ is a forgery of message m . The resulting forgery is both passive and selective, which is the most severe type. Therefore, the intractability of the semi-logarithm problem is a necessary condition for the security of ECDSA.

A semi-logarithm may be regarded as an ECDSA signature of some message whose hash is one (without actually supplying the message). Thus, at first glance, the semi-logarithm might not seem to be significantly different than the forging of a ECDSA signature. But semi-logarithms do not depend on a hash function, unlike ECDSA. Therefore, the semi-logarithm problem is formally different from the problem of forging of a ECDSA signature. One reason for considering the semi-logarithm problem is to isolate the role of the hash function and the group in analyzing the security of the ECDSA.

Intractable Discrete Logarithm : For a group $\langle G \rangle$, the (discrete) logarithm of a group element P to the base G is the integer x such that $P = [x]G$. Finding the discrete logarithm of P allows one to find its semi-logarithm, via

$$(t, u) = (f([k]G), k^{-1}(1 + td)),$$

therefore allowing the forging of ECDSA signatures. The forger is both passive and selective. Indeed, this forger recovers the elliptic curve private key, which might potentially result in yet greater damage than mere forgery, if, say, the key is used for other purposes.

Almost-Bijective Conversion Function : The conversion function f is α -clustered if some element t^* of its range has a large preimage of size at least α times the domain size. If f is not α -clustered, then it is said to be almost bijective of strength $1/\alpha$. An α -clustered conversion function means that random (t^*, u) are semi-logarithms with probability at least α . Thus, an average of about $1/\alpha$ tries are needed to obtain a semi-logarithm.

Unguessable and Arithmetically Unbiased Private Key Generation :

Clearly, if the generator for the static private key x is guessable, in the sense that an adversary can guess its values fairly easily, then passive selective forgery is possible. Guessability, sometimes called min-entropy, is measured by the maximum probability of any value of x . If the ephemeral private key k is guessable, then active selective forgery is possible, since the private key x is determined from k and a signature (r, s) by the formula $x = r^{-1}(ks - H(m)) \pmod{q}$.

Furthermore, a series of attacks has been published that show if the random number generator used for k exhibits certain types of bias, the private key x can be obtained. If k ever repeats for different messages m and m' , then the private key may be solved from the two respective signatures (r, s) and (r', s') by $x = (se' - s'e)/(s'r - sr') \pmod{q}$. Bellare, Goldwasser and Micciancio [21] showed that if a linear congruential random number generator were used for k , then the private key could also be found; Smart and Howgrave-Graham [174] and Nguyen and Shparlinski [261, 262] both showed that if bits of k are known, due to partial key exposure, then x can be found using lattice theory; and Bleichenbacher [31] showed that if k is biased towards a certain interval, then x can be recovered with a larger but feasible amount of work. Such biases result in active selective forgery, because the forger uses a signing oracle to find the private key and can then sign any message it wants.

Properly Implemented Range Checking : If an implementation does not check that $r \neq 0$, then the following forgery is possible. The forger needs to select the EC domain parameters in such a way that $G = [t]Z$, where Z is a group element satisfying $f(Z) = 0$ and $t \in \mathbb{Z}$. For the ECDSA conversion function, such points Z , if they exist, can be found as follows. Let x have the binary representation of qu for some integer u and try to solve for the appropriate y such that (x, y) lies on the curve (here x is not to be confused with the private key). Repeat until a point is found or until all legal values of x are exhausted. Most of the NIST recommended curves have such points Z . The forged signature is $(0, t^{-1}H(m))$.

This forgery is the severest kind: passive selective. Two limitations mitigate its severity, however. First, an implementation error is needed. Hence, non-repudiation is not totally defeated because a trusted third party can use a correct implementation to resolve the signature validity. Accordingly, the

owner of the key Y should never be held liable for such signatures. The second limitation is that the forgery is a domain parameter attack. Usually, a trusted third-party authority generates and distributes domain parameters, including G . The attack presumes a corrupt authority. Note that a verifiably random generation of G , which the revision of [ANSI X9.62] will allow, prevents this attack without relying on the implementer to check that $r \neq 0$.

Rarely Zero Hash : If the *effective* hash function, which is the raw hash truncated and reduced modulo q , has probability p of equalling 0, then passive selective forgery is possible, as follows. The forger chooses signature $(r, s) = (f([t]Y), t^{-1}r)$, for some $t \in \mathbb{Z}$. If the selected message is a zero of the hash, which happens with probability p , then the forged signature is valid because

$$f([s^{-1}]([H(m)]G + [r]Y)) = f([tr^{-1}]([0]G + [r]Y)) = f([t]Y) = r.$$

This and other conditions on the hash function refer to the *effective* hash function. This qualification is important for the security analysis because the reduction modulo q might cause the condition to fail if q was chosen by an adversary. If the adversary chose the elliptic curve domain parameters, then it is possible that q was chosen as the output, or the difference between two outputs, of the unreduced hash function, which would permit the adversary to find a zero or a collision in the effective (reduced) hash function.

Notice that clustering at values other than zero does not necessarily lead to a passive existential forgery. It can lead to other kind of attacks, as outlined below, because clustering at certain values can lead to weakened second-preimage resistance of the hash function.

Zero-Resistant Hash : A zero finder of a hash function is a probabilistic algorithm that finds a message m such that $H(m) = 0$. A hash function is *zero-resistant* if no zero finder exists. A passive existential forger can be constructed from a zero finder in a similar manner to above. The forger chooses signature $(r, s) = (f([t]Y), t^{-1}r)$ and, using the zero finder, finds a message m such that $H(m) = 0$. Then (r, s) is a valid signature on m . Note that the forger is only existential because the forger has no control on the m found by the zero finder.

A zero-resistant hash function is clearly rarely zero. The converse is false: a rarely zero hash function can fail to be zero-resistant. Note that this strict separation of the properties is also reflected in the types of forgery they relate to, so therefore it is important to consider both properties in a security analysis.

A special case of this attack was first described by Vaudenay [331] as a domain parameter attack on DSA, where the zero is found by choosing q for m such that $H(m) \equiv 0 \pmod{q}$.

First-Preimage Resistant (One-Way) Hash : An inverter of a hash function is a probabilistic algorithm that, if given a random hash value e , finds a message m such that $H(m) = e$. An inverter can be used to build a passive existential forger using a technique similar to that for a zero finder. The forged signature is $(r, s) = (f([g]G + [y]Y), ry^{-1})$, and the forged message m is a preimage of $e = gs$. The forger is existential, not selective, because the forger has no control on the m found by the inverter.

If the hash function does not have an inverter, it is *preimage-resistant*. Such a hash is also known as a *one-way hash function*. We have just shown that for the signature scheme to be secure, the effective hash must be one-way.

Preimage and zero-resistance are essentially independent properties of hashes. In particular, both properties are necessary to resist passive existential forgery.

Second-Preimage Resistant Hash : A second-preimage finder of a hash function is a probabilistic algorithm that, if given a random message m from a distribution U , a second message m' is found such that $H(m') = H(m)$. A second-preimage finder can be used to build an active selective forger. The forger obtains a signature (r, s) of m' from the signing oracle and then outputs (r, s) as a forgery of m . The forger is selective because it can forge a challenge message m , and it is active because it requires access to a signing oracle.

Note that a second-preimage finder can be constructed from a hash-inverter provided that the distribution U has an image distribution $H(U)$ that covers most of the range of H with sufficiently high probability. This is implied by results noted by Stinson [321]. Therefore, under these conditions, a second-preimage finder also implies the existence of passive selective forger.

Collision-Resistant (One-Way) Hash : A collision finder of a hash function is a probabilistic algorithm that finds two messages m and m' such that $H(m') = H(m)$. A collision finder results in an active existential forger as follows. The forger obtains a signature (r, s) of m' from the signing oracle and then outputs (r, s) as a forgery of m , where m and m' have been obtained from the collision finder. This forger is existential because it has no control over the messages m and m' found by the collision finder, and it is active because it queries a signing oracle.

Note that a collision finder can be constructed from a second-preimage finder, provided that the distribution U is easy to sample. Also, generally, a collision finder can be constructed from a zero finder, unless the zero finder always outputs one particular zero of the hash function.

II.2.3. Extra Conditions and Models for Sufficiency. The conditions and models discussed below are used in the hypotheses of various provable security results for ECDSA. These conditions and models are not known to be necessary for the security of ECDSA. As such, they represent the other

end of the gap in what is known about the security of ECDSA. Unlike for the necessary conditions discussed above, it is plausible to argue that these extra conditions and models are so strong that the security results have little value. Indeed, the ideal would be to not use any conditions or models other than the known necessary conditions. For each condition or model below, one could try to prove the condition necessary, or to find a better proof that relies on strictly weaker conditions or models.

Almost-Invertible Conversion Function : A conversion function f is *almost-invertible* if it has an *almost inverse* g , which is an efficient probabilistic algorithm that behaves like an inverse function to f in the following sense. The algorithm g takes input $z \in_R \mathbb{Z}/q\mathbb{Z}$ and generates output $P \in \langle G \rangle \cup \{\text{Invalid}\}$ such that:

1. One has $P \neq \text{Invalid}$ with probability at least $\frac{1}{10}$ as assessed over random choices made for z and g .
2. If $P \neq \text{Invalid}$, then $f(P) = z$.
3. If independently random inputs $z \in_R \mathbb{Z}/q\mathbb{Z}$ are repeatedly given to g until the output $P \neq \text{Invalid}$, the resulting probability distribution of such P is indistinguishable from the distribution of random $P \in_R \langle G \rangle$.

In other words, g will successfully find a preimage of its input at least one time in ten, and, furthermore, the resulting preimage has no (computationally) detectable bias.

Clearly almost-invertibility implies almost-bijectivity. In particular, it is a strict strengthening of a necessary condition. When used as part of the hypothesis of a provable security result, it implies this particular necessary condition, almost-bijectivity.

Adaptive Semi-Logarithm Resistance : For a conversion function f , group $\langle G \rangle$, and $e \in \mathbb{Z}/q\mathbb{Z}$ with $e \neq 1$, an e -shifted semi-logarithm of a group element P to the base G is a pair of integers (t, u) such that $t = f([u^{-1}]([e]G + [t]P))$. The adaptive semi-logarithm problem is to find a semi-logarithm of a point P to the base G given access to an oracle for e -shifted semi-logarithms. Adaptive semi-logarithm resistance means that the adaptive semi-logarithm problem is intractable.

The adaptive semi-logarithm resembles an active forgery of ECDSA, except that no hash functions are involved. The necessity of adaptive semi-logarithm resistance has not yet been established. Clearly, adaptive semi-logarithm resistance is a strengthening of semi-logarithm resistance, a known necessary condition.

Pseudo-Random Random Number Generator : The security proofs generally assume that the static key x and the ephemeral private keys k , if

any, are generated uniformly and independently at random from the private key space $\mathbb{Z}/q\mathbb{Z}^*$.

In practice, this means that the private keys can be generated with a pseudo-random number generator, which is something whose output is indistinguishable from private keys generated truly randomly as above. Indeed, if the schemes were broken when instantiated with such a pseudo-random number generator, then the pseudo-random number generator can be distinguished from a true random number generator as follows. Run the forgery attack on ECDSA instantiated by the challenge number generator. If the forgery attack succeeds, then the challenge is likely to come from a pseudo-random generator; if the forgery attack fails, then the challenge number generator is likely to be truly random.

Obviously, when considering passive forgery, the generator for ephemeral keys is never invoked, so there is no condition on how k is generated. The generation of x is of course still vital.

If the signatures are generated in a deterministic mode, meaning that k is a secret deterministic function of the message being signed, then pseudo-randomness only applies on a per-message basis, not a per-signature basis. Some of the proofs really only apply in this deterministic mode.

Uniform (Smooth) Hash Function : Let $H : \{0, 1\}^* \rightarrow \mathbb{Z}/q\mathbb{Z}$ be the effective hash function. Let $U \subseteq \{0, 1\}^*$ be such that

1. For $m \in_R U$, $e = H(m)$ can be efficiently generated.
2. For each $e \in \mathbb{Z}/q\mathbb{Z}$, the set $P_e = h^{-1}(e) \cap U$ is sufficiently large so that, even if P_e is known, the probability $1/|P_e|$ is sufficiently small to make guessing a randomly selected secret element of P_e infeasible.

Let $b \in_R \{1, 2\}$. A probabilistic algorithm D_h is an (ϵ_D, τ_D, U) -*distinguisher* for h if D_h accepts as input $e \in_R \mathbb{Z}/q\mathbb{Z}$ if $b = 1$ or $e = H(m)$ for $m \in_R U$ if $b = 2$, and if D_h outputs, in running-time at most τ_D a guess $d \in \{1, 2\}$. Further, $d = b$ with probability at least $\frac{1}{2} + \epsilon_D$ assessed over the random choices of b , e and D_h . If no such D_h exists, then h is *uniform* or *smooth* of strength (ϵ_D, τ_D, U) .

Uniformity (or smoothness) can be exhibited by some very simple “hash” functions that are neither collision resistant nor one-way. For example, the function

$$H : \begin{cases} \{0, 1\}^* & \rightarrow & \mathbb{Z}/q\mathbb{Z}, \\ x & \mapsto & \bar{x} \pmod{q}, \end{cases}$$

where $\bar{x}$ is the integer represented by the bit-string x , is uniform for $U = \{x : 0 \leq x < q^2\}$. Thus, it is highly plausible that the effective hash function for ECDSA is also very uniform for an appropriate choice of U if H is built from a cryptographic hash function.

A uniform, collision resistant hash is one-way. To see this directly, suppose there was an inverter for a uniform hash function H . Select a random

message $m \in_R U$ and compute $e = H(m)$. Give e to the hash-inverter. The hash-inverter outputs some message m' , such that $H(m') = e = H(m)$. Thus, (m', m) is a collision, unless $m' = m$. But it is information-theoretically infeasible to find m from e alone, so $m' \neq m$ and a collision is found. Moreover, because e is indistinguishable from random elements in $\mathbb{Z}/q\mathbb{Z}$, the hash-inverter's probability cannot be affected by the choice of e or else it would be a distinguisher. Thus the collision finder has the same success rate and efficiency as the hash-inverter. Thus, when our security results use collision resistance and uniformity as hypotheses, one-wayness is an implicit additional hypothesis [321].

Ideal Group Model : The ideal group model, or generic (group) model was introduced into cryptography by Shoup [303]. A version of this model was applied to ECDSA by Brown [52]. In the ideal group model, the group operation is accessed only via an oracle, which accepts queries of two elements that will be subtracted. The representations of group elements are random elements of a subset. The representations may be thought of as encryptions. Any practically implementable group is not an ideal group, since a practical implementation must perform group operations without invoking an oracle.

When the group is modelled as ideal, the adversary's queries to and responses from the oracle can be analyzed in the following way. Some queries will be group elements that have not occurred in earlier queries or responses. These elements can be regarded as independent. All other elements can be regarded as integer combinations of the independent elements. So long as distinct integer combinations represent distinct elements, the oracle has not revealed any relations between the independent elements; otherwise, a dependency will have been discovered. Since the independent elements are effectively random elements in the group, it is possible to formulate the probability of a dependency being discovered, and it is small if the number of queries is small. By convention, the base point G and public key Y are independent. If the number of queries is small, then the probability that a dependency such as $Y = [x]G$, equivalent to finding the private key, could be discovered is small. When a group is modelled as ideal, it is possible to prove, as Shoup did [303], that certain problems related to the group (such as the discrete logarithm problem) cannot be solved with less than a certain number of queries.

Algorithms to solve problems in a group that work for any group, including an ideal group, are called generic. Conversely, algorithms to solve problems in a specific group that do not work for other groups, including an ideal group, are called specific. Generic algorithms for solving the discrete logarithm problem cannot be much faster than the square root of the largest prime factor in group order [303], whereas specific algorithms might potentially work faster for particular groups. A provable security result in the

generic group model only protects against adversaries that are generic with respect to the group. Adversaries to ECDSA specific to elliptic curve groups are thus not ruled out.

More theoretically, Dent [100] has shown it is also possible that a given cryptographic scheme with a provable security result in the ideal group model has the property that there exists a successful adversary against the scheme, for all efficiently implementable groups. This adversary is not a generic group adversary because it is limited to efficiently implementable groups. This limitation, though, involves the creation of contrived schemes designed to allow such adversaries, and thus it is generally believed to be only a theoretical limitation.

Ideal Hash Model : Also known as the random oracle model or ideal hash paradigm, the ideal hash model is analogous to the ideal group model. More precisely, it models the hash function by a random oracle function. Any security proof in the ideal hash model should be regarded as only a validation or argument of security with respect to the design of the scheme, irrespective of the hash function. Moreover, it suffers from the same severe limitations as the ideal group model, as shown by Canetti, Goldreich and Halevi [54].

II.3. Provable Security Results

The following provable security results are in the opposite direction of the results about the necessity of certain conditions on the primitive components of ECDSA.

THEOREM II.7. *If the effective hash function is rarely zero and the semi-logarithm resistance holds for the group and conversion function, then ECDSA has passive selective unforgeability.*

The conditions in the hypothesis of this result are also necessary conditions. The passive selective unforgeability of ECDSA is equivalent to these conditions. The main way this result could be improved is to determine whether semi-logarithm resistance is equivalent to or strictly weaker than discrete logarithm resistance.

THEOREM II.8. *If the hash function is zero-resistant and weakly collision resistant (second-preimage resistant), the conversion function and group are together adaptive semi-logarithm resistant, and the random number generator is pseudo-random (indistinguishable from random), then ECDSA has active selective unforgeability.*

THEOREM II.9. *If the hash function is zero-resistant and one-way, the conversion function is almost invertible, and the group is modelled as ideal (i.e., as a generic group), then ECDSA has passive existential unforgeability.*

Another result for passive existential unforgeability is a special case of the next result below for active existential unforgeability. The conditions of this special case are no stronger or weaker than the conditions of the result above.

THEOREM II.10. *If the hash function is idealized as a random oracle, then semi-logarithm resistance implies that ECDSA has active existential unforgeability.*

Because semi-logarithm resistance is a necessary condition, the result is tight — modulo the random oracle model for the hash function. Using a random oracle allows us to weaken the adaptive semi-logarithm resistance used in the result for active selective unforgeability.

THEOREM II.11. *If the elliptic curve group is modelled as a generic group, then almost invertibility of the conversion function f , and smoothness, collision resistance and zero resistance of the hash function h together imply that ECDSA has active existential unforgeability.*

Results in both directions are summarized in Table II.2, which should help illustrate the gap between the various necessary and sufficient conditions. The first column indicates the type of security in terms of passiveness and selectiveness. The second column gives the direction of the result; $\Rightarrow$ to show that security implies certain conditions and $\Leftarrow$ to show that security is implied by certain conditions. The remaining columns are for the underlying components of ECDSA. Rows with $\Rightarrow$ list all the conditions necessary, and rows with $\Leftarrow$ list all the conditions (or models) that together are sufficient.

Notice that the strongest unforgeability type (corresponding to the least harmful, i.e., an active existential forger) has two results complementary in the sense of using different idealized models, one idealizing the hash (ROM) and the other idealizing the group (GEN). The only other security type that uses an idealized model (GEN) is the passive existential. The proof of active existential unforgeability using the idealized hash model specializes to passive existential unforgeability, so this type of security also has complementary idealizations in provable security results.

II.4. Proof Sketches

II.4.1. Passive Selective Unforgeability. Suppose F is a passive selective forger and the hash function H is rarely zero. Then we will find a semi-logarithm to the base G at a random challenge point P as follows. Run F on a random selected message m and public key $Y = [H(m)]P$ to get a forgery (r, s) . Then $(r, H(m)^{-1}s)$ is the desired semi-logarithm.

II.4.2. Active Selective Unforgeability. Suppose F is an active selective forger. Then we will either solve the adaptive semi-logarithm problem to the base G at a random challenge point P , or find a zero or second preimage of a random challenge message m for the hash function H , as follows. Run

TABLE II.2. Summary of All Results

Security		h	f	$\langle G \rangle$	k
SEL PAS	$\Rightarrow$	RZ	SLR		—
	$\Leftarrow$	RZ	SLR		—
SEL ACT	$\Rightarrow$	ZFR+WCR	SLR		NAB
	$\Leftarrow$	ZFR+WCR	ASLR		PR
EXI PAS	$\Rightarrow$	ZFR+OW	SLR		—
	$\Leftarrow$	ZFR+OW	AI	GEN	—
EXI ACT	$\Rightarrow$	ZFR+CR	SLR		NAB
	$\Leftarrow$	ROM	SLR		PR
	$\Leftarrow$	ZFR+CR+S	AI	GEN	PR

PAS	Passive	SLR	Semi-logarithm resistant
ACT	Active	ZFR	Zero-resistant
SEL	Selective	CR	Collision-resistant
EXI	Existential	GEN	Generic group
OW	One-way	ASLR	Adaptive semi-logarithm resistant
RZ	Rarely zero	ROM	Random oracle hash
PR	Pseudo-random	AI	Almost invertible
S	Smooth	WCR	Weakly collision-resistant

F on a random selected message m and public key $Y = [H(m)]P$. Answer signature queries $m_i \neq m$ by querying the shifted semi-logarithm oracle on $e_i = H(m)^{-1}H(m_i)$ if $e_i \neq 1$. If some $e_i = 1$, then stop, having found a second preimage m_i of m . Otherwise, F produces a forgery (r, s) on the message m . Then $(r, H(m)^{-1}s)$ is the desired semi-logarithm.

II.4.3. Passive Existential Unforgeability. Suppose F is a passive existential forger. Then we will be able to invert the hash function H at a random challenge $e \in \mathbb{Z}/q\mathbb{Z}$ as follows. Run F against a modified generic group oracle. The generic group oracle is modified in that a random preselected oracle response is designated to take the value $f^{-1}(ey/g)$, where g and y are the integers such that the forger can observe that the output of the group query is $[g]G + [y]Y$ (queries not of this form are not selected for modification). Because e is random and f is almost-invertible, the modified response will be effectively random in $\mathbb{Z}/q\mathbb{Z}$, and therefore the success rate of F will remain unaffected. If Q is the number of group queries used by F , then $1/Q$ is the chance that message forgery will satisfy $s^{-1}H(m) = g$ and $s^{-1}r = y$, and in particular, m is a preimage of e .

II.4.4. Active Existential Unforgeability with Idealized Hash. Suppose F is an active existential forger. Then we will use F to find the semi-logarithm to the base G of a challenge point P , as follows. Run F against

a random oracle hash, modified as follows. A random preselected oracle response is designated to take a value e chosen at random from $\mathbb{Z}/q\mathbb{Z}$. Because e is random, the modified response will be effectively random in $\mathbb{Z}/q\mathbb{Z}$, and therefore the success rate of F will remain unaffected. Run the active existential forger with challenge public key $Y = [e]P$. If h is the number of hash queries used by F , then $1/h$ is the chance that message forgery will satisfy $H(m) = e$, in which case we have the semi-logarithm $(r, s/e)$, where (r, s) is the signature.

In order to handle signature queries, a further modification to the random oracle hash is used. To answer signature queries for a message m_i , just choose a random pair $(x_i, y_i) \in \mathbb{Z}_q^2$ and compute $r_i = f([x_i]G + [y_i]Y)$, $s_i = r_i/y_i$ and $e_i = x_i r_i / y_i$. If the forger queries m_i to the hash oracle later in the run, the oracle responds with e_i . To ensure that signature queries can be answered after hash queries, answer every hash query m_i using the method above. The hash query responses e_i are effectively random and therefore the random oracle hash is indistinguishable from a true random oracle, from the forger's perspective.

Nevertheless, the modified query responses do have a distinguishable impact on the signature query responses because these become deterministic. This phenomenon also applies to the forking lemma approach. Thus, arguably this security proof only appears to apply to the deterministic mode of ECDSA. Indeed, intuitive randomized signatures do potentially leak more information about the private key than deterministic signatures, because an adversary might somehow be able to exploit multiple different signatures of a single message to compute information about the private key.

II.4.5. Active Existential Unforgeability with Idealized Group. Suppose that F is an active existential forger, that the hash function H is zero resistant and smooth, and that the conversion function is almost invertible. We will use F to find a collision in the hash function H . Run F against a modified generic group oracle. The generic group oracle is modified in that each oracle response is designated to take the value $f^{-1}(H(\widetilde{m}_i y / g))$, where $\widetilde{m}_i$ is chosen at random from U where g and y are the integers such that the forger can observe that the output of the group query is $[g]G + [y]Y$ (queries not of this form are not selected for modification). Because $H(\widetilde{m}_i)$ is indistinguishable from random and f is almost-invertible, the modified response will be effectively random in $\mathbb{Z}/q\mathbb{Z}$, and therefore the success rate of F will remain unaffected. Furthermore, because H is smooth, F will not be able to guess $\widetilde{m}_i$. Therefore the forger's signature queries m_i and forged message m are distinct from the messages $\widetilde{m}_i$. In order for the signature to verify with chance better than random, it would need to have one of the queries involving $\widetilde{m}_i$ and therefore $H(m) = H(\widetilde{m}_i)$, which is the collision desired.

II.5. Further Discussion

II.5.1. Semi-Logarithms Versus Discrete Logarithms. The discrete logarithm problem is traditionally considered the primary basis for the security of ECDSA. But the semi-logarithm problem is considered in this chapter because it is not obvious whether it is equivalent to or weaker than the discrete logarithm. The security of ECDSA depends on this potentially weaker problem. The security of ECDSA will be clarified once the relationship between these two problems is established.

If it is ever proved that ECDSA is secure if the elliptic curve discrete logarithm is intractable, then this would establish the equivalence of the elliptic curve semi-logarithm and discrete logarithm problems. This is true even if the security proof uses the random oracle model for the hash function. Also, elliptic curve experts can study the equivalence of these two problems without looking at ECDSA or hash functions.

II.5.2. Applying the Results to ECDSA Variants. Of course, a number of variants of ECDSA are possible, and it is worthwhile to consider briefly whether the results apply to these variants.

Hashing kG : One suggested variant of ECDSA includes the ephemeral public key $[k]G$ as part of the input to the message digesting hash function. In the random oracle model idealizing for hash functions, this adds to the provable security because the active existential unforgeability of ECDSA can be proved relying on the discrete logarithm problem, not the semi-logarithm. On the other hand, under different hypotheses this variation diminishes the provable security, as follows. Some security proofs for ECDSA no longer apply. The proof of active selective unforgeability that idealizes neither hash nor group, for example, does not apply. The obstacle to these proof techniques seems to result from the difficulty of separating the roles of the group and the hash in the variant. Furthermore, this also exemplifies the principle that design complexity generally hinders provable security.

DSA and One-Way Conversion Functions : Almost-invertibility does not hold for the DSA conversion function. Indeed, in DSA, the conversion function is probably a good one-way function, which is quite the opposite of almost-invertible. Therefore, the provable security results using almost-invertibility of the conversion function do not apply well to DSA. Therefore, DSA and ECDSA have different provable security properties. In particular, they are not as analogous as at first they seem.

One-wayness, however, would intuitively seem to add security. The situation seems paradoxical. One explanation is that almost-invertible means the function is similar to the trivial identity function, and the identity function

is very simple to analyze. A more complex design might be more secure, although it could also be more complex to analyze. With added complexity, one can never discount the possible appearance of an attack. For DSA, it's possible that somebody could attack the conversion function. For example, DSA could be insecure because the conversion function used is not almost-bijective or for some other reason. One could assume that the DSA conversion function is almost-bijective and try to find a provable security result, but nobody has done this yet.

The intuition that a one-way conversion function imparts some kind of security attribute is not entirely ungrounded. Almost-invertibility means that the public key can be recovered from the message and signature (with reasonable probability). A one-way conversion function seems to prevent this. This difference does not have an impact on GMR security. It could have other impacts such as anonymity (hiding the signer's identity) or efficiency (omitting the public key). Hiding the public key is not a stated objective of ECDSA.

Non-Pseudorandom k : No result has shown that k needs to be indistinguishable from a uniform random integer in $[1, q - 1]$. Indeed, since ECDSA is not meant to provide confidentiality, the need for indistinguishability is not clear. Intuitively, a weaker condition than pseudo-randomness ought to be sufficient for ECDSA. Certainly, the private keys must be unguessable and arithmetically unbiased, because of known attacks, but these are weaker conditions than pseudo-randomness.

To see why pseudo-randomness might not be necessary for k , consider the following. Choose truly random private keys k subject to the condition that their hashes display a given pattern. Such k fail to be pseudo-random because they can be distinguished by applying the hash function, yet they do not seem to be weak. They are unlikely to have an attackable arithmetic bias. They may have enough entropy to be unguessable.

Also, some of the results do not involve a signing oracle and therefore do not require the ephemeral private keys k to be generated pseudo-randomly.

Deterministic k : In some of the proofs, the signing oracle value has the property that the same message query always gives the same signature response. Technically, this means the proof is only applicable to the deterministic mode of ECDSA signing, where k is chosen as a secret deterministic function of the message m being signed. An intuitive explanation that the deterministic mode is more secure is that it reveals less signatures and therefore less information about the private key. A very cautious implementation of ECDSA could use the deterministic mode so that these provable security results apply.

II.5.3. Attack-Like Attributes of ECDSA. Despite the proofs of GMR security of ECDSA, it might be argued that GMR security itself is not the “right” definition. Logically speaking, of course, a definition, by definition, cannot be right or wrong. Nonetheless, cryptography is a practical science, not a purely mathematical one, and therefore definitions ought to be tailored to pertinent concerns, not purely arbitrary ones. With this perspective, some alternative definitions of security for signatures in which ECDSA can be deemed “insecure” are explored and assessed for their pertinence.

Many of the attributes that we explore hinge on the particular conversion function f used in ECDSA. Altering f to avoid these attributes could potentially do more harm than good, diminishing the reputational security of ECDSA and the provable security of ECDSA. Accordingly, addressing these attributes is best handled through other means.

Signature Non-Anonymity : Given a valid ECDSA signature (r, s) on message m , the associated public key Y can be recovered, as follows. (Note that this does not violate the GMR definition of signature security.) Solve for the public key as $Y = [r^{-1}][s]R - [H(m)]G$, where R is selected from $f^{-1}(r)$, the set of points in the preimage of r .

Self-Signed Signatures : A signature of a message is self-signed if the message contains the signature. A self-signed ECDSA signature can be generated as follows. Choose random k and s . Compute $r = f([k]G)$. Form the message m containing the signature (r, s) . Compute $e = H(m)$. Now solve for a private key x that makes this signature valid, which can be found as $x = (\pm sk - e)/r \pmod{q}$.

This attribute does not violate GMR security. Indeed, it may be a useful attribute in the sense that it can be used to ensure that the private key was not stolen. It may also be useful for server-assisted key generation, where a server adds entropy to the message m so the signer’s private key x has enough entropy. Additional modifications to the self-signed signature verification are necessary, however, if the server cannot be trusted and the signer’s entropy for k is weak.

Unknown Private Key : A valid ECDSA signature can be generated without knowing the private key and yet not violate the GMR definition of signature security, as follows. This can be done for any elliptic curve domain parameters and any message m , by first generating a random value of the signature (r, s) and then solving for the public key as $Y = [r^{-1}][s]R - [H(m)]G$, where $R \in f^{-1}(r)$, the set of points in the preimage of r . If $f^{-1}(r) = \{\}$, then just try another value of r .

This attribute does not defeat GMR security because it does not attack a target public key. This “attack” could be thwarted if f were made one-way, but then the provable security offered by the proofs assuming almost-invertibility of f would be lost.

Invalid Public Key : An ECDSA signature that can be verified can be generated for as many messages as desired after committing beforehand to a public key Y , as follows. Commit to an invalid public key Y of small order a (for this work, the verifier is to use Y without validating it). Given message m , generate random s and t . Compute $r = f([H(m)]G + [t]Y)$. If $r = t \pmod{a}$, then (r, s) will be valid.

This attribute does not defeat GMR security because it is the adversary that chooses the public key, as in the previous example, and moreover, the public key is invalid. Notwithstanding that this attribute is not a real attack against a signer, the verifier is well advised to validate all public keys, because it would be rather foolish to rely on invalid public keys.

Duplicate Signatures : An ECDSA signature that is valid for two given messages m_1 and m_2 can be generated, as follows. Choose random k , then compute $R = [k]G$, $r = f(R)$, $x = (2r)^{-1}(H(m_1) + H(m_2)) \pmod{q}$ and $Y = [x]G$ and $s = k^{-1}(H(m_1) + xr) \pmod{q}$. This works because $f(P) = f(-P)$.

This attribute of ECDSA does not defeat GMR security because it requires generation of the public key, as in both the previous examples. One can argue duplicate signatures qualify as an “attack” defeating non-repudiation. Then one would conclude that GMR security does not lead to non-repudiation.

On the other hand, to repudiate a signature of a message m_2 , on the grounds that it is a duplicate, is to argue that signer intended to sign m_1 and that some adversary discovered the duplication with m_2 . If an adversary could do this, then GMR security would be defeated, because the signer did not sign m_2 . So the adversary’s repudiation argument reduces to claiming that ECDSA is not GMR-secure. Therefore the break between GMR security and non-repudiation has not been broken.

This particular duplicate signature attack is of even less of a concern than a general duplicate signature attack, because it exposes the private key as $x = (2r)^{-1}(H(m_1) + H(m_2)) \pmod{q}$. The hypothetical third-party forger could therefore compute the private key and discard the duplicate signature in favour of full-fledged forgeries, drastically weakening the credibility of the alleged repudiator. Also, exposure of the private key almost certainly shows that the signer intentionally created the private key solely in order to fraudulently repudiate the signature.

Malleable Signatures : A valid ECDSA signature that was never generated by the signer with public key Y can be generated given any message m ,

as follows. Get the signer to generate a signature (r, s) of the message m , and then generate $(r, -s \pmod{q})$ as the desired signature. This works because $f(P) = f(-P)$, as in the previous example.

This attribute does not defeat GMR security because the message m has already been signed by the signer. A forgery, under the GMR definition, must be of a message that was never signed, which excludes m .

This attribute has been called malleability, but perhaps benign malleability is more apt, because both terms have precedents in the context of encryption schemes, and this attribute is much more akin to the benign variety, where the malleability does not impinge on the message itself but merely to the cryptographic data. Arguably, benign malleability is always present in that a given piece of data often has multiple encodings, and any non-malleable scheme can be transformed into a benignly malleable one by applying such an encoding.

CHAPTER III

Proofs of Security for ECIES

A.W. Dent

Provable security in an encryption setting is very similar to provable security in a digital signature setting (see Chapter II). In both cases we aim to make meaningful, mathematically rigorous statements about the security of cryptosystems and provide proofs that these statements are correct.

Generally, a security proof attempts to show how difficult “breaking” a cryptographic scheme is, in terms of how difficult it is to solve some mathematical problem. If we can show that the difference between breaking the cryptographic scheme and solving the underlying mathematical problem is only small, and we assume that solving the underlying problem is difficult to do, then we can have some measure of assurance in the security of the cryptographic scheme. The main difference between proving security in the signature setting and in the encryption setting is deciding what is meant by “breaking” the scheme.

Before we launch into the complicated and arcane world of provable security, it is useful to take a moment to consider its history. The field of provable security for public-key encryption schemes has a history almost as long as public-key encryption itself. The most significant early papers on provable security are by Rabin in 1979 [279] and Goldwasser and Micali in 1984 [149]. These papers proposed schemes with provable security properties based on the properties of modular arithmetic – obviously this was too early for elliptic curves! For many years after this, provable security remained the province of the academic. It seemed that most practical cryptosystems were too complicated for a formal proof of security to be found. The schemes that did have proofs of security were often thousands of times slower than the ones that were used in practice. It took two revolutionary steps to bring provable security back into the province of the practical cryptographer. The first involved setting out the correct model for an attack. This was essentially done by Rackoff and Simon in 1991 [280]. The second was the formalization of the random oracle model by Bellare and Rogaway in 1993 [24]. The random oracle model involves assuming that any hash functions are ideal, i.e., act as totally random functions, and it allows for a much easier analysis of a scheme.

Throughout this chapter we will be considering the ECIES encryption scheme; see Section I.4. ECIES provides a very good example of the advantages and disadvantages of the provable security approach to cryptography.

The scheme was one of the first formalized versions of a hybrid encryption scheme, that is, an asymmetric encryption scheme that uses both asymmetric and symmetric cryptographic techniques. Essentially, ECIES uses an elliptic curve Diffie–Hellman key transport to generate a random symmetric key, which is used to encrypt and MAC a message.

The idea of hybrid encryption had been folklore amongst the cryptographic community for years but had never been formalized. This led to some weak implementations of the ideas (see, for example, the attack of Boneh, Joux and Nguyen [41]). ECIES was introduced as a concrete example of a hybrid encryption scheme, and it came with a security proof that guaranteed that, providing the scheme was used in the right way, the scheme could not be broken. On top of this, because the idea of hybrid encryption had been known to the cryptographic community for so long, the scheme also had a high level of reputational security. It is not surprising then that it quickly found its way into many mainstream cryptographic products and standards.

III.1. Definitions and Preliminaries

III.1.1. Security for Encryption. To be able to make any meaningful mathematical statements about the security of an encryption scheme, we first have to be very clear about what constitutes an encryption scheme.

DEFINITION III.1. *A public-key encryption scheme is a triple of algorithms $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ where*

- $\mathcal{G}$ is a probabilistic algorithm called the key generation algorithm. It takes no input (except randomness) and outputs a key-pair (pk, sk) . Here pk is the public key and must be distributed to everyone who wishes to use the encryption algorithm, and sk is the private key and should only be made available to those parties who have permission to decrypt messages.
- $\mathcal{E}$ is a probabilistic algorithm called the encryption algorithm. It takes as input the public key pk and a message m drawn from some message space $\mathcal{M}$ defined by the public key. It outputs a ciphertext C in some ciphertext space $\mathcal{C}$, also defined by the public key.
- $\mathcal{D}$ is a deterministic algorithm called the decryption algorithm. It takes as input the private key sk and a ciphertext C from the ciphertext space $\mathcal{C}$ and it returns either a message $m \in \mathcal{M}$ or an error symbol $\perp$.

A public-key encryption scheme is *sound* if for all valid key-pairs (pk, sk) we have that $\mathcal{D}(\mathcal{E}(m, pk), sk) = m$ for every message $m \in \mathcal{M}$. If $\mathcal{E}$ is a deterministic algorithm, then $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is said to be a deterministic encryption scheme (even though $\mathcal{G}$ is still a probabilistic algorithm).

An attacker for a public-key encryption scheme is a pair of probabilistic algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. The precise nature of the attacker depends upon two factors: (1) what the attacker is trying to do, and (2) what access the

attacker has to the scheme. These days we only really consider attackers who are trying to do one of two things: either an attacker is trying to invert a ciphertext to find the message it represents (the one-way game) or they are trying to tell which of two messages a ciphertext is the encryption of (the indistinguishability game).

In both cases the attacker plays a game against a challenger. The challenger isn't a real person or even a computer program, it is simply a convenient way of describing the way in which the attacker's inputs are constructed. Take, for example, the one-way game. Here a challenger picks a message uniformly at random from the set $\mathcal{M}$ of all possible messages, encrypts it, and gives the resulting ciphertext to the attacker to attempt to decrypt. In reality the "challenger" does not exist – who would implement a cryptosystem and include a program that just encrypts random messages for no apparent purpose? – it is simply a convenient way for us to explain the precise mathematical model that we used to assess the security of the scheme.

DEFINITION III.2. *The one-way (OW) game for an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ consists of four major steps:*

1. *A challenger generates a random key-pair (pk, sk) by running the key generation algorithm $\mathcal{G}$.*
2. *The attacker runs $\mathcal{A}_1$ on the input pk . It returns some state information s .*
3. *The challenger chooses a message m uniformly at random from the message space $\mathcal{M}$. It computes the challenge ciphertext $C^* = \mathcal{E}(m, pk)$.*
4. *The attacker runs $\mathcal{A}_2$ on the input (C^*, pk, s) . It returns a guess m' for m .*

The attacker wins the game if $m' = m$.

The one-way game is a fairly weak notion of security. There are many schemes that are secure against attackers playing the one-way game that are still not secure enough to be used in practice. The indistinguishability game is a much stronger notion of security.

In the indistinguishability game the attacker is asked to provide two messages (usually termed m_0 and m_1). The challenger picks one of these messages at random and encrypts it, giving the resulting ciphertext C^* back to the attacker. The attacker then has to guess which message the challenger encrypted. This may initially seem to be quite easy for the attacker to do: surely the attacker can just encrypt both of the messages and compare their encryptions to C^* ? Well, of course the attacker can do this but this may not help. We must remember here that the encryption algorithm may be probabilistic, which means that if the same message is encrypted twice we are unlikely to get the same ciphertext both times, and knowing one encryption of a message may not help us recognise another encryption of the same message.

DEFINITION III.3. *The indistinguishability (IND) game for an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ consists of four major steps:*

1. *A challenger generates a random key-pair (pk, sk) by running the key generation algorithm $\mathcal{G}$.*
2. *The attacker runs $\mathcal{A}_1$ on the input pk . It returns two messages m_0 and m_1 , as well as some state information s .*
3. *The challenger chooses a bit $\sigma \in \{0, 1\}$ uniformly at random. It computes the challenge ciphertext $C^* = \mathcal{E}(m_\sigma, pk)$.*
4. *The attacker runs $\mathcal{A}_2$ on the input (C^*, pk, s) . It returns a guess σ' for σ .*

The attacker wins the game if $\sigma' = \sigma$.

The advantage of the attacker $\mathcal{A}$ in playing the IND game is defined to be

$$Adv_{\mathcal{A}} = |\Pr[\sigma' = \sigma] - 1/2|.$$

The advantage of an attacker is a measure of how much better the attacker $\mathcal{A}$ is than the attacker who simply guesses σ' at random. An attacker who guesses σ' uniformly at random has a success probability of $1/2$ and an advantage of 0. It should be noted that some authors prefer to define the advantage of an attacker to be twice this value so that it is nicely scaled as a value between 0 and 1.

Another aspect that frequently causes some confusion is the state information s . The state information s that $\mathcal{A}_1$ passes to $\mathcal{A}_2$ can contain any information that may be of help to $\mathcal{A}_2$. This could include the messages m_0 and m_1 or information about the way in which $\mathcal{A}_1$ chose them, information about the decryptions of certain messages or any other information that $\mathcal{A}_1$ thinks may be of use to $\mathcal{A}_2$ in winning the game. It is nothing mysterious, it is simply a way of making sure that $\mathcal{A}_2$ knows what $\mathcal{A}_1$ was doing when it chose m_0 and m_1 .

The indistinguishability game is a strong notion of security. Suppose there was some algorithm $\mathcal{A}$ that could, given a ciphertext, deduce whether the decryption of that ciphertext would pass or fail some test T , the test T could be something like whether the last bit of the message is 0 or 1, or whether the sum of the bits in the message is even or odd. We could then construct an attacker that could win the indistinguishability game by choosing the two messages m_0 and m_1 such that m_0 passes the test T and m_1 fails the test T . The attacker could then tell if the challenge ciphertext is an encryption of m_0 or m_1 by running the algorithm $\mathcal{A}$ to see if the decryption of the challenge ciphertext would pass or fail the test T .

This means that if a cryptosystem is secure against attackers playing the indistinguishability game, then an attacker can gain no meaningful results (i.e., any information about whether it would pass or fail any kind of test) about a message from its encryption.

Next we must consider what access an attacker has to the scheme. Obviously the attacker must have access to a description of the algorithms in the scheme and the public key pk . This means that an attacker will be able to encrypt messages for himself. However, just as we allowed an attacker to request the signatures of certain messages when considering the security of digital signatures, we may allow an attacker to decrypt certain ciphertexts of his choice. Obviously care must be taken to make sure that the attacker isn't allowed to decrypt the challenge ciphertext!

An attacker requests the decryption of a ciphertext C by outputting C and then entering a special freeze state. The attacker is then given the response to his query and re-started. In this way we can think of an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ as consisting of two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ that each work in the same way as the forger for a digital signature (see Definition II.3). The first algorithm $\mathcal{A}_1$ runs in multiple rounds, each consisting of two plays, the first made by the challenger and the second by the attacker. In Round 0, the challenger starts by generating a valid key-pair (pk, sk) . $\mathcal{A}_1$ is then given the public key pk and some predetermined initial state X_0 . It then outputs a state X_1 and either a request for the decryption of a ciphertext C or, in the case of the indistinguishability game, two messages (m_0, m_1) . In the case of the one-way game, $\mathcal{A}_1$ halts without any extra output when it terminates. If $\mathcal{A}_1$ has requested the decryption of a ciphertext, then the challenger computes the decryption m of C using the private key sk and re-starts $\mathcal{A}_1$ with state X_1 and input m . The second algorithm $\mathcal{A}_2$ works similarly but initially takes as input the challenge ciphertext C^* and an initial state X_i that was given by $\mathcal{A}_1$ in its final round (this is equivalent to the state information s). When $\mathcal{A}_2$ terminates it must output its guess for either σ (in the indistinguishability game) or m (in the one-way game).

Whilst this is one way of thinking about an attacker's ability to request decryptions, it will suit our purposes to be a little more relaxed and merely to think of an attacker having the ability to ask a god-like oracle for them. If an attacker is allowed to request decryptions, then it is said to have access to a decryption oracle, and these requests are assumed to take a single unit of time to answer (such is the power of the oracle).

There are three models that are used to characterise an attacker's access to a decryption oracle.

DEFINITION III.4. *Consider an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$.*

If the attacker has no access to a decryption oracle, then it is said to be running a chosen plaintext attack (CPA), because it has the ability to choose which plaintexts (messages) it wishes to encrypt. Remember, it knows the public key and can therefore encrypt any message it wants to. Here the attacker cannot request the decryption of any ciphertext.

If the first algorithm $\mathcal{A}_1$ has access to a decryption oracle (i.e., can request the decryption of ciphertexts) but the second algorithm $\mathcal{A}_2$ has no access to a

decryption oracle then the attacker is said to be running a chosen ciphertext attack (CCA1).¹

If both the first algorithm $\mathcal{A}_1$ and the second algorithm $\mathcal{A}_2$ have access to a decryption oracle, then the attacker is said to be running an adaptive chosen ciphertext attack (CCA2). In this case we must assume that the second algorithm only has access to an imperfect decryption oracle that will not decrypt the challenge ciphertext C^* . This is the strongest notion of access that we will consider here.

It is easy to see that if an attacker can break a scheme using a chosen plaintext (CPA) attack, then there is an attacker who can break that scheme with a chosen ciphertext (CCA1) attack. Hence, if we can show that a scheme is secure against attackers running CCA1 attacks, then we can be sure that that scheme is also secure against attackers running CPA attacks. Similarly, a scheme that resists attacks made by CCA2 attackers is also resistant to CCA1 attackers and CPA attackers.

To completely define the level of security that an algorithm has we must look at the best attackers in a certain model, i.e., we must specify whether the attacker is playing the indistinguishability game or the one-way game and what kind of access the attacker has. We often abbreviate these definitions to their initials. For example, attackers playing the indistinguishability game using adaptive chosen ciphertext attacks are often referred to as IND-CCA2 attackers and are said to be making an IND-CCA2 attack.

These days, it is generally agreed that a public-key encryption scheme is only secure if it resists attackers making IND-CCA2 attacks, and it is in its resistance to this kind of attack that we will be examining ECIES. More information about attack models can be found in [20].

III.1.2. Underlying Mathematical Problems. Up to now cryptographers have been unable to find security proofs that prove the security of an algorithm directly (without the need to rely on any assumptions or simplifications). There are several good reasons for this. The first has to do with the very nature of public-key cryptography. The way public-key cryptography works means that it is impossible to find a proof of security in which the attacker has access to infinite computational resources – given the public key of an algorithm, there can only be one corresponding private key, and so an attacker with unlimited time and computational resources could check each possible private key in turn until he finds the correct one (thus breaking the

¹Attacks that use CCA1 access to a decryption oracle are also sometimes known as *midnight* attacks or *lunchtime* attacks because it is the sort of attack that could be used by an intruder who breaks into an office either at night or whilst the proper occupant is at lunch. If the intruder was unable to recover the occupant's private key directly, then he might try and run some kind of program that makes use of the proper occupant's decryption rights and that would henceforth allow the intruder to decrypt messages meant for the occupant without having to know his private key.

scheme). Therefore, if we wish to make any meaningful statement about the security of a scheme, we must place some kind of bound on the computational power of the attacker.

Even with this restriction we still run into problems. Any efficient proof of security that guaranteed the security of an algorithm in concrete terms would most likely prove the complexity-theoretic conjecture that $P \neq NP$. Whilst this is widely believed to be true, and despite many years of study, a proof has not yet been found. Any such proof would be a major step forward in the field of complexity theory. Therefore the best that one can reasonably hope for from a proof of security is that it relates the security of a scheme to the difficulty of solving some kind of underlying mathematical problem that is believed to be difficult to solve. In other words, even though we use the term “security proof”, our actual level of faith in the security of a scheme depends upon how difficult we *believe* the underlying problem is.

There are three major underlying problems that have traditionally been used in proving the security of elliptic curve based cryptosystems. These are all based on the reputational security of the ECDH protocol (see Section I.3). The three traditional underlying problems are the *computational Diffie–Hellman problem (CDH)*, the *decisional Diffie–Hellman problem (DDH)* and the *gap Diffie–Hellman problem (gap DH)*.

DEFINITION III.5. *Let G be a cyclic group with prime order $\#G$ (and with the group action written additively), and let P be a generator for G .*

The computational Diffie–Hellman problem is the problem of finding $[ab]P$ when given $([a]P, [b]P)$. We assume that a and b are chosen uniformly at random from the set $\{1, \dots, \#G - 1\}$.

The decisional Diffie–Hellman problem is the problem of deciding whether $[c]P = [ab]P$ when given $([a]P, [b]P, [c]P)$. We assume that a and b are chosen uniformly at random from the set $\{1, \dots, \#G - 1\}$ and c is either equal to ab (with probability $1/2$) or chosen uniformly at random from the set $\{1, \dots, \#G - 1\}$ (with probability $1/2$). The advantage that an algorithm A has in solving the DDH problem is equal to

$$|\Pr[A \text{ correctly solves the DDH problem}] - 1/2|.$$

The gap Diffie–Hellman problem is the problem of solving the CDH problem when there exists an efficient algorithm that solves the decisional Diffie–Hellman problem on G . In other words, the gap Diffie–Hellman problem is the problem of finding $[ab]P$ when given $([a]P, [b]P)$ and access to an oracle that returns 1 when given a triple $([a]P, [\beta]P, [\alpha\beta]P)$ and 0 otherwise. We assume that a and b are chosen uniformly at random from the set $\{1, \dots, \#G - 1\}$.

Any algorithm that can solve the CDH problem on a group can also be used to solve the gap DH problem and the DDH problem. Hence it is better to try and show that a scheme can only be broken if the CDH problem is easy to solve, as this automatically tells us that the scheme can only be broken if

both the gap DH and DDH problems are easy to solve too. The assumption that the CDH problem is hard to solve is called the *CDH assumption*. The gap DH assumption and the DDH assumption are defined in the same way.

The difficulty of solving the CDH problem on an elliptic curve group is the same as the difficulty in breaking the ECDH protocol on that elliptic curve when the attacker has only seen one execution of that protocol. The ECDH protocol has been around for so long that it is trusted to be secure even though there exists no formal proof of security. There have been some results which suggest that the difficulty of breaking the ECDH protocol is related to the difficulty of solving the elliptic curve discrete logarithm problem, which is considered hard to solve in most elliptic curve groups (see [ECC, Chapter V]).

It is interesting that the security of almost all elliptic curve based cryptosystems has been based on the difficulty of breaking the Diffie–Hellman protocol because the Diffie–Hellman protocol is not restricted to elliptic curve groups: it can be applied to any cyclic group (although the resulting protocol may not be secure). Up until very recently there have been no underlying mathematical problems that have been used as a basis for a proof of security and which are specific to elliptic curves alone. Recently, however, cryptosystems have been developed that are based on certain properties of the Weil and Tate pairings. This will be explained more thoroughly in Chapter IX and Chapter X.

III.1.3. Security for Symmetric Cryptography. One of the problems with developing a security proof for ECIES is that it relies on undefined symmetric encryption and MAC schemes. Obviously, if the symmetric encryption scheme is weak and it is possible to derive information about the message from its symmetric encryption, then no amount of elliptic curve cryptography is going to make the ECIES encryption of the message secure. That the MAC algorithm is of equal importance takes a little more thought and is best illustrated by means of an example.

Suppose that the symmetric encryption scheme used in an instantiation of ECIES is a Vernam cipher. In a Vernam cipher a fixed-length message is XORed with a key of equal size in order to create a ciphertext. Decryption of a ciphertext is given by once again XORing the ciphertext with the key to give the original message. Suppose further that we remove the MAC algorithm from the ECIES scheme. This scheme is now definitely insecure against CCA2 attacks. Suppose that the challenge ciphertext $C^* = (U^*, c^*)$ is the ECIES encryption of a message m^* . (How this message is chosen and whether the attacker is playing the IND or OW game is irrelevant here.) This means that $c^* = m^* \oplus k^*$, where k^* is some symmetric key derived from the secret key x and U^* . An attacker can recover k^* by asking for the decryption of a ciphertext $C = (U^*, c)$, where $c \neq c^*$. If the decryption of C is m , then the attacker knows that $k^* = c \oplus m$, and he can easily recover m^* from c^* and k^* .

However, this attack is not possible when we use a good MAC algorithm within ECIES as this makes it infeasible for an attacker to find the MAC r of the symmetric ciphertext c without knowing the MAC key. Hence the attacker cannot find a ciphertext (U^*, c, r) that will be decrypted by the decryption algorithm and therefore cannot recover k^* . The use of a MAC scheme essentially means that the symmetric encryption algorithm only needs to resist passive attacks, where the attacker does not have the ability to encrypt and decrypt messages himself. It allows symmetric ciphers that are normally considered too weak for practical purposes, such as the Vernam cipher, to be used within ECIES (although readers who are considering implementing ECIES with a Vernam cipher are referred to Section III.3.3 for a cautionary tale).

Formally we define a symmetric encryption scheme to be a pair of deterministic algorithms (ENC, DEC) with the property that for any message $m \in \mathcal{M}$ and any bit-string K of a predetermined length we have

$$\text{DEC}(\text{ENC}(m, K), K) = m.$$

In defining the security of a symmetric encryption scheme, we consider its resistance to attackers $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that are playing a game which is very similar to the IND game we defined for attackers against asymmetric encryption schemes. This game consists of four major steps:

1. A challenger generates a key K totally at random from the set of all possible keys.
2. The attacker runs $\mathcal{A}_1$. It returns two messages m_0 and m_1 of equal size, as well as some state information s .
3. The challenger chooses a bit $\sigma \in \{0, 1\}$ uniformly at random. It computes the challenge ciphertext $C^* = \text{ENC}(m_\sigma, K)$.
4. The attacker runs $\mathcal{A}_2$ on the input (C^*, s) . It returns a guess σ' for σ .

The attacker wins the game if $\sigma' = \sigma$. The advantage of an attacker in playing this game is defined to be

$$|\Pr[\sigma' = \sigma] - 1/2|.$$

This definition is taken from [93] and is a weaker form of the “Find-then-Guess” notion of security contained in [19]. Note that the attacker is not allowed to request the encryption or decryption of any message – this is called a *passive* attack.

It is fairly easy to see that the Vernam cipher satisfies this definition of security; indeed, in this model the Vernam cipher is perfect.

The formal definition of the security of a MAC scheme is more like the security definitions associated with a digital signature scheme. A MAC scheme is a deterministic algorithm MAC that, for any message m and any fixed-length bit-string K , produces a fixed-length output $r = \text{MAC}(m, K)$. For our purposes an attacker is a probabilistic algorithm $\mathcal{A}$ that attempts to find

a new MAC for any message. We test an attacker's strength by computing its success probability in playing the following game:

1. A challenger generates a key K uniformly at random from the set of all possible keys.
2. The attacker runs $\mathcal{A}$. It may, at any time, query an oracle with a message m . If so, the challenger will compute $r = \text{MAC}(m, K)$ and return r to the attacker.
3. $\mathcal{A}$ eventually returns a pair (m', r') .

The attacker wins the game if $\mathcal{A}$ has not queried the oracle to find the MAC of m' , i.e., $\mathcal{A}$ has not submitted m' to the oracle in step 2, and $r' = \text{MAC}(m', K)$.

III.2. Security Proofs for ECIES

It has become generally accepted that a general-purpose public-key encryption scheme should be resistant to attacks made by IND-CCA2 attackers, i.e., attackers that are playing the indistinguishability game using adaptive chosen ciphertext attacks, so we will only consider the security of ECIES against this kind of attack.

There has never been a simple proof of the security of ECIES. Given that it uses undefined symmetric encryption and MAC schemes, not to mention an undefined key derivation function (the security of which we have not yet even considered), it is not really surprising that no simple proof has been found. In this section we will sketch three proofs of security for ECIES; each one is forced to make some assumption about the way ECIES works in order to make the security analysis easier.

III.2.1. Using a Non-Standard Assumption. The original proof of security for ECIES, which was given by Abdalla, Bellare and Rogaway [1], was based on a non-standard underlying mathematical problem. By this we mean that it was not based on the difficulty of solving either the CDH problem, the DDH problem or the gap DH problem. The authors instead chose to make an assumption based not only on the difficulty of breaking the Diffie–Hellman protocol but also on the nature of the key derivation function used. They called this the *hash Diffie–Hellman problem*.

DEFINITION III.6. Let G be a cyclic group with prime order $\#G$ (and with the group action written additively), and let P be a generator of G . Furthermore, let KD be a key derivation function and let l be a fixed parameter.

The hash Diffie–Hellman problem is the problem of deciding whether $\alpha = KD([a]P, l)$ when given $([a]P, [b]P, \alpha)$. We assume that a and b are chosen uniformly at random from the set $\{1, \dots, \#G - 1\}$ and that α is either equal to $KD([ab]P, l)$ (with probability $1/2$) or chosen uniformly at random from the set of all l -bit strings (with probability $1/2$). To help the attacker, it will be given access to a hash Diffie–Hellman oracle: an oracle that, when given

a group element U , will return $KD([b]U, l)$. The attacker will not be allowed to query this oracle with the input $[a]P$.

The advantage that an algorithm $\mathcal{A}$ has in solving the hash Diffie–Hellman problem is equal to

$$|\Pr[\mathcal{A} \text{ correctly solves the hash Diffie–Hellman problem}] - 1/2|.$$

This definition is fairly sneaky. It is not actually a single definition but a family of definitions that depend upon your choice of a key derivation function. The hash Diffie–Hellman problem may indeed be hard to solve for many key derivation functions (and choices of l), but it is also possible that, for a particular choice of key derivation function and l , the hash Diffie–Hellman problem is easy to solve. In many ways, adopting the hash Diffie–Hellman problem as the basis for a security proof only changes the problem from proving the security of the cryptosystem to proving that the hash Diffie–Hellman problem is hard to solve for your chosen key derivation function.

The security proof for ECIES in this model shows the following:

RESULT III.7. *Suppose $\mathcal{A}$ is an IND-CCA2 attacker that breaks ECIES with a “significant” advantage, runs in a “reasonable” amount of time and only makes a “reasonable” number of queries to the decryption oracle. Then either*

- *there exists an algorithm $\mathcal{B}$ that solves the hash Diffie–Hellman problem with a significant advantage, runs in a reasonable amount of time and only makes a reasonable number of queries to the hash Diffie–Hellman oracle;*
- *there exists an attacker $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2)$ that breaks the symmetric cipher with a significant advantage and runs in a reasonable amount of time; or*
- *there exists an attacker $\mathcal{F}$ that breaks the MAC scheme with a significant probability and runs in a reasonable amount of time.*

The terms “significant” and “reasonable” all have precise technical definitions, but it will be unnecessary to consider these in detail. It will suffice to think of “significant” as meaning “large enough to be useful” and “reasonable” as “not too large”.

We will now sketch the proof.

Suppose we have an IND-CCA2 attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that breaks ECIES with a significant advantage, runs in a reasonable amount of time and only makes a reasonable number of queries to the decryption oracle.

We begin by defining an algorithm $\mathcal{B}$ that attempts to solve the hash Diffie–Hellman problem. At the start of its execution, $\mathcal{B}$ is presented with a triple $([a]P, [b]P, \alpha)$. The algorithm $\mathcal{B}$ runs in several stages:

1. $\mathcal{B}$ sets the public key $Y = [b]P$ and runs $\mathcal{A}_1$ on the input Y until it outputs two messages, m_0 and m_1 , and some state information s .
2. $\mathcal{B}$ picks a bit $\sigma \in \{0, 1\}$ uniformly at random and sets $(k_1 || k_2) = \alpha$. It forms the challenge ciphertext $C^* = ([a]P, c^*, r^*)$, where $c^* =$

$\text{ENC}(m_\sigma, k_1)$ and $r^* = \text{MAC}(c^*, k_2)$. Note that if $\alpha = KD([ab]P, l)$, then C^* is a correct encryption of the message m_σ .

3. Next, $\mathcal{B}$ runs $\mathcal{A}_2$ on the input (C^*, Y, s) . $\mathcal{A}_2$ returns a guess σ' for σ .
4. If $\sigma' = \sigma$, then $\mathcal{B}$ returns **true** (i.e., that $\alpha = KD([ab]P, l)$), otherwise $\mathcal{B}$ returns **false**.

The idea behind the proof is that if $\alpha = KD([ab]P, l)$, then the attacker $\mathcal{A}$ will be attacking a valid instance of ECIES and so will have a significant advantage in guessing σ correctly. If $\alpha \neq KD([ab]P, l)$, then α is completely random. This means that the message m_σ has been encrypted and the MAC of the ciphertext computed with random keys. If this is the case, then one of two things will happen: either (1) the attacker $\mathcal{A}$ will still be able to break the scheme and recover σ with a significant probability, in which case the attacker doesn't really care about the keys and must be attacking the symmetric part of the scheme, or (2) the attacker can no longer break the scheme.

Suppose that an attacker $\mathcal{A}$'s advantage in breaking the scheme is significantly reduced when $\alpha \neq KD([ab]P, l)$. In this case $\mathcal{A}$ is less likely to guess σ correctly if random keys are used to encrypt m_σ and to create the MAC of the ciphertext, which in turn means that $\mathcal{B}$ is more likely to output **true** if $\alpha = KD([ab]P, l)$ than if it doesn't. In other words, $\mathcal{B}$ has a significant advantage in breaking the hash Diffie–Hellman problem. So, if we assume that solving the hash Diffie–Hellman problem is hard to do, then any attacker that is trying to break the scheme must still be able to do so when performing the symmetric encryption and MAC using random keys.

The only problem that remains with the algorithm $\mathcal{B}$ is that it has to be able to respond to the decryption queries that $\mathcal{A}$ makes. Normally these would be answered by a decryption oracle, but since $\mathcal{B}$ has no access to a decryption oracle, it must answer these queries itself. Fortunately this is fairly easy to do. If $\mathcal{A}$ asks for the decryption of a ciphertext (U, c, r) where $U \neq [a]P$, then we may find the relevant symmetric keys $k_1 || k_2$ by querying the hash Diffie–Hellman oracle on the input U ; then we may decrypt c and r as normal. If $\mathcal{A}$ asks for the decryption of a ciphertext $([a]P, c, r)$ with $(c, r) \neq (c^*, r^*)$, then we can use the symmetric keys $(k_1 || k_2) = \alpha$ to decrypt c and r as normal. Of course there is a possibility that $\mathcal{A}_1$ might request the decryption of the challenge ciphertext C^* (remember that $\mathcal{A}_2$ is not allowed to request the decryption of C^*), but, in order to do this, $\mathcal{A}_1$ would have to guess that the challenge ciphertext would contain $[a]P$. Since $[a]P$ is chosen at random from all the non-identity points of $\langle P \rangle$, and because $\mathcal{A}$ is only allowed to make a reasonable number of decryption queries, this will happen with such a small probability that we may cheerfully ignore it.

As we mentioned earlier, it is, of course, possible that the attacker $\mathcal{A}$ gained his advantage in breaking ECIES not by determining some information about the symmetric keys but by attacking the symmetric part of the algorithm without knowing those keys. In that case the advantage of the

algorithm $\mathcal{B}$ would not be significant. However, then the attacker would still be able to break the scheme if the keys used by the symmetric encryption algorithm and the MAC algorithm were completely random.

The idea that we can replace the key used to encrypt the challenge ciphertext with a random key means that we can construct an algorithm $\mathcal{C}$ that breaks the symmetric cipher. Remember that here $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2)$ is a two-stage algorithm that has access to a decryption oracle for the symmetric cipher with an unknown, randomly selected symmetric key and is trying to decide which of two messages (m_0 and m_1) a given symmetric ciphertext is an encryption of (see Section III.1.3). Algorithm $\mathcal{C}$ works by using the symmetric cipher as part of ECIES and using $\mathcal{A}$ to break it. The first algorithm $\mathcal{C}_1$ works as follows:

1. $\mathcal{C}_1$ picks a private key x uniformly at random from the set $\{1, \dots, q-1\}$ and sets the public key Y to be $[x]P$.
2. $\mathcal{C}_1$ runs $\mathcal{A}_1$ on the input Y . $\mathcal{A}_1$ will return two messages m_0 and m_1 , as well as some state information s .
3. $\mathcal{C}_1$ returns the two messages, m_0 and m_1 , and some state information $s' = (s, x)$.

The challenger then picks a bit $\sigma \in \{0, 1\}$ uniformly at random and computes $c^* = \text{ENC}(m_\sigma, k_1)$, where k_1 is some random (unknown) secret key that the challenger chose at the start. The attacker then runs $\mathcal{C}_2$ on the input (c^*, s') . Algorithm $\mathcal{C}_2$ runs as follows.

1. $\mathcal{C}_2$ takes s' and recovers the secret key x and $\mathcal{A}$'s state information s .
2. $\mathcal{C}_2$ chooses a random point U^* in $\langle P \rangle$.
3. $\mathcal{C}_2$ chooses a random MAC key k_2 and computes $r^* = \text{MAC}(c^*, k_2)$.
4. The challenge ciphertext for $\mathcal{A}$ is set to be $C^* = (U^*, c^*, r^*)$.
5. $\mathcal{C}_2$ runs $\mathcal{A}_2$ on the input (C^*, Y, s) . $\mathcal{A}_2$ outputs a guess σ' for σ .
6. $\mathcal{C}_2$ outputs σ' .

Algorithm $\mathcal{C}$ makes the problem of breaking the symmetric cipher look exactly like the problem of breaking ECIES, except for the fact that the challenge ciphertext was computed using randomly generated symmetric keys rather than from the keys given by $(k_1 || k_2) = KD([x]U^*, l)$. We have already shown that if $\mathcal{B}$'s advantage in solving the hash Diffie-Hellman problem is not significant, then $\mathcal{A}$ should still be able to break ECIES in this case. Hence $\mathcal{C}$ should have a significant advantage in breaking the symmetric cipher.

Again the only problem we haven't explained is how to respond to the decryption requests $\mathcal{A}$ makes. Once again, this is fairly simple. Since $\mathcal{C}$ knows the private key x , it can easily decrypt any ciphertext (U, c, r) where $U \neq U^*$ in the normal way. If $\mathcal{A}$ requests a decryption of the form (U^*, c, r) , then $\mathcal{C}$ simply responds with ‘‘invalid ciphertext’’. It does this because it does not wish to confuse $\mathcal{A}$. All decryption requests for ciphertexts of the form (U^*, c, r) should use the same symmetric key to decrypt c as was used to encrypt the challenge ciphertext c^* – but it does not know what this is!

Hence it is easier for $\mathcal{C}$ to just ignore any requests to decrypt ciphertexts of the form (U^*, c, r) than to try and guess what the answer should be.

Obviously this means there is a chance that $\mathcal{C}$ will respond to a ciphertext query (U^*, c, r) by saying “invalid ciphertext” when the ciphertext is valid and should be decrypted. However, then we must have that $c \neq c^*$, as otherwise $r = r^*$ and $\mathcal{A}$ is querying the decryption oracle on the challenge ciphertext, which would mean that $\mathcal{C}$ has succeeded in forging the MAC pair (c, r) . We can use this to build an attacker $\mathcal{F}$ that breaks the MAC scheme. If $\mathcal{F}$ does not have a significant success probability, then we know that all of the decryption oracle responses that $\mathcal{A}$ is given by $\mathcal{C}$ are correct and so $\mathcal{C}$ should have a significant advantage in breaking the symmetric scheme.

III.2.2. Using an Idealized Key Derivation Function. If one is uneasy about the validity of a security proof based on a non-standard and unstudied problem (like the hash Diffie–Hellman problem), then one can find a proof of security for ECIES based on the gap Diffie–Hellman problem if one is prepared to accept the use of the random oracle methodology [24].

The random oracle methodology is a model in which the key derivation function is modelled as being perfect, i.e., as a completely random function. The attacker, who needs to be able to compute the value of the key derivation function, is given access to this random function by means of an oracle that will evaluate it for him. Unfortunately this model has been shown to have some theoretical weaknesses [54]. Nevertheless security proofs constructed using the random oracle methodology (in the “random oracle model”) are still considered a very good *heuristic* guide to the security of a cryptosystem.

The security proof for ECIES in the random oracle model shows the following:

RESULT III.8. *Suppose $\mathcal{A}$ is an IND-CCA2 attacker that breaks ECIES with a significant advantage in the random oracle model, and that $\mathcal{A}$ runs in a reasonable amount of time and only makes a reasonable number of queries to the decryption oracle. Then either*

- *there exists an algorithm $\mathcal{B}$ that solves the gap Diffie–Hellman problem with a significant probability, runs in a reasonable amount of time and only makes a reasonable number of queries to the decisional Diffie–Hellman oracle;*
- *there exists an attacker $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2)$ that breaks the symmetric cipher with a significant advantage and runs in a reasonable amount of time;*
or
- *there exists an attacker $\mathcal{F}$ that breaks the MAC scheme with a significant probability and runs in a reasonable amount of time.*

For the most part the proof is very similar to the proof given in Section III.2.1. The only difference is in how one constructs the algorithm $\mathcal{B}$.

We now sketch the proof.

Suppose we have an IND-CCA2 attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ that breaks ECIES with a significant advantage, runs in a reasonable amount of time and only makes a reasonable number of decryption queries.

At the start of its execution, the algorithm $\mathcal{B}$, which is attempting to solve the gap Diffie–Hellman problem, is presented with the pair $([a]P, [b]P)$. The algorithm $\mathcal{B}$ runs in several stages:

1. $\mathcal{B}$ sets the public key Y to be $[b]P$ and runs $\mathcal{A}_1$ on the input Y until it outputs two messages, m_0 and m_1 , plus some state information s .
2. $\mathcal{B}$ picks a bit σ uniformly at random from the set $\{0, 1\}$ and symmetric keys k_1 and k_2 uniformly at random from the appropriate key spaces. It forms the challenge ciphertext $C^* = ([a]P, c^*, r^*)$, where $c^* = \text{MAC}(m_\sigma, k_1)$ and $r^* = \text{MAC}(c^*, k_2)$.
3. Next, $\mathcal{B}$ runs $\mathcal{A}_2$ on the input (C^*, Y, s) . It returns a guess σ' for σ .
4. Since we are using the random oracle model, the attacker $\mathcal{A}$ cannot evaluate the key derivation function itself and must instead ask $\mathcal{B}$ to evaluate it. Hence $\mathcal{B}$ knows all the elliptic curve points on which $\mathcal{A}$ has queried the key derivation function. $\mathcal{B}$ searches through all these points until it finds one, Q say, such that $Q = [ab]P$. Note $\mathcal{B}$ can use the DDH oracle to check this as $([a]P, [b]P, Q)$ will be a correct solution to the DDH problem.
5. If such a point Q exists, then $\mathcal{B}$ returns Q ; otherwise, $\mathcal{B}$ returns a point picked at random.

The idea behind the proof is that if $\mathcal{A}$ has not queried the key derivation function on $[ab]P$, then it has no clue about the symmetric keys used to encrypt the challenge ciphertext. Because the key derivation function is totally random, querying the key derivation function on other points will not help you gain any information about $KD([ab]P, l)$. Hence if the attacker $\mathcal{A}$ has not queried the key derivation function on $[ab]P$, then we might as well be using completely random symmetric keys (which, in fact, $\mathcal{B}$ is doing) and the attack of $\mathcal{A}$ will work just as well. On the other hand, if the attacker $\mathcal{A}$ does query the key derivation function on the input $[ab]P$, then $\mathcal{B}$ will have solved the gap Diffie–Hellman problem.

This essentially puts us in the same situation as last time: either $\mathcal{B}$ has a significant probability of solving the gap Diffie–Hellman problem or $\mathcal{A}$'s attack will work even if we use random keys. The rest of the proof (constructing $\mathcal{C}$ and $\mathcal{F}$) is exactly as in the previous section.

Unfortunately we are not quite finished. The devil of this proof is in the details of how $\mathcal{B}$ answers $\mathcal{A}$'s oracle queries. Remember that $\mathcal{A}$ can now ask two different types of oracle queries. It can request the decryption of some ciphertext or it can request that the key derivation function be evaluated on some input. Furthermore, the two different types of query have to be consistent with each other: if $\mathcal{A}$ requests the decryption of a ciphertext (U, c, r) , then we must make sure that the symmetric keys we use to decrypt c and r are

the same symmetric keys that $\mathcal{A}$ would get if we queried the key derivation oracle on $[b]U$ (and don't forget that $\mathcal{B}$ doesn't know the value of b !).

We get around this by making a long list of the queries that $\mathcal{A}$ makes. Each entry on the list is a triple (U, T, K) , where U and T are elliptic curve points:

- U is the elliptic curve point contained in the ciphertext.
- T is the input to the key derivation function, so T should equal $[b]U$.
- K is the output of the key derivation function, i.e., the keys to be used with symmetric encryption and MAC schemes.

If $\mathcal{A}$ requests the decryption of a ciphertext (U, c, r) , then $\mathcal{B}$ responds in the following way:

1. If there is an entry of the form (U, T, K) or $(U, -, K)$ on the query list, then we use the symmetric keys $K = (k_1 || k_2)$ to decrypt c and r and return the correct message.
2. Otherwise we check all the entries of the form $(-, T, K)$ to see if $T = [b]U$. We can use the DDH oracle to do this, since if $T = [b]U$, then (U, Y, T) is a solution to the DDH problem. If such an entry $(-, T, K)$ is found, then we replace it with the entry (U, T, K) and decrypt c and r using the symmetric keys $K = (k_1 || k_2)$.
3. If neither of the above two approaches works, then we have no information at all about the symmetric keys that should be used. Therefore we can randomly generate $K = (k_1 || k_2)$ (as the keys are the output of a totally random function) and use this to decrypt c and r . We then add $(U, -, K)$ to the query list.

Similarly, if $\mathcal{A}$ requests that the key derivation function be evaluated on the input T , then $\mathcal{B}$ responds in the following way:

1. If there is an entry of the form (U, T, K) or $(-, T, K)$ on the query list, then we return K .
2. Otherwise we check all entries of the form $(U, -, K)$ to see if $T = [b]U$, as before, we can use the DDH oracle to do this. If such an entry is found, then we replace it with the entry (U, T, K) and return K .
3. If neither of the above two approaches works, then we can randomly generate K , add $(-, T, K)$ to the query list and return K .

Essentially what we have done here is shown that if the gap Diffie–Hellman problem is difficult to solve and the key derivation function can be modelled as a random oracle, then the hash Diffie–Hellman problem is difficult to solve too. This is why this proof is so similar to the last one.

III.2.3. Using an Idealized Group. Whilst the previous two proofs made assumptions about the properties of the key derivation function in order to simplify the security analysis of ECIES, the proof we will discuss in this section makes assumptions about the group over which ECIES is defined. We will assume that the group is ideal, i.e., that the representations of group

elements as bit-strings has no relation to the structure of the group. This means that any attacker who wishes to break the scheme cannot take advantage of any special arithmetic properties of the group on which the scheme is defined. This idea is the same as the one used to prove the security of ECDSA in Section II.4.4; however, the way in which we use this idea will be slightly different.

Specifically, we will be using the version of the generic group model proposed by Shoup [303]. Here we assume that the group elements are not represented as elliptic curve points but as randomly generated bit-strings of a given length. Since the group elements are now represented by strings of random bits, an attacker cannot take advantage of any of the usual properties of elliptic curves when attacking the scheme. Unfortunately the attacker can not add or subtract group elements either, so we are forced to give him access to an oracle that will perform these operations for him.

This may all seem a bit contradictory. After all, ECIES is explicitly defined over an elliptic curve group and an attack that takes advantage of the arithmetic properties of the elliptic curve group to break ECIES is still a valid attack against the cryptosystem. The aim of a proof of security in the generic group model is to provide evidence that any attack against the scheme would have to exploit some special feature of elliptic curve groups. Since finding such a special feature would be a major step forward in elliptic curve cryptography, we can have some measure of assurance in the security of ECIES. Unfortunately it has been shown that, as with the random oracle model, the generic group model has some theoretical weaknesses [100]. Nevertheless, just as with the random oracle model, security proofs constructed in this model are still considered a very good heuristic guide to the security of a cryptosystem.

The proof of security of ECIES in the generic group model was given by Smart [310] and shows the following:

RESULT III.9. *Suppose $\mathcal{A}$ is an IND-CCA2 attacker that breaks ECIES with a significant advantage in the generic group model, and that $\mathcal{A}$ runs in a reasonable amount of time and only makes a reasonable number of queries to the decryption oracle. Then either*

- *there exists an algorithm $\mathcal{B}$ that solves the decisional Diffie–Hellman problem with a significant advantage and runs in a reasonable amount of time;*
- *there exists an attacker $\mathcal{C} = (\mathcal{C}_1, \mathcal{C}_2)$ that breaks the symmetric cipher with a significant advantage and runs in a reasonable amount of time;*
or
- *there exists an attacker $\mathcal{F}$ that breaks the MAC scheme with a significant probability and runs in a reasonable amount of time.*

This result is strengthened by the following result of Shoup [303]:

RESULT III.10. *In the generic group model, there exist no algorithms that solve the decisional Diffie–Hellman problem on a cyclic group of large prime order that have both a significant advantage and run in a reasonable amount of time.*

The proof of Result III.9 is very similar to the previous proofs and is therefore omitted.

III.3. Other Attacks Against ECIES

One of the problems with security proofs is that they tend to cause a certain level of complacency about using a cipher. The truth is that a security proof is only valid in a very precise set of circumstances. It assumes that the attacker has access to no other information besides the public key and (possibly) a decryption oracle, and has a perfect implementation of the cryptosystem. In the real world an attacker may have access to much more information than the theoretical attacker did in the security proof. Attacks that make use of information or abilities that are not available to an attacker in a security proof are often called side-channel attacks.

Some side-channel attacks are only relevant to one particular cryptosystem, but others can be used to attack whole classes of ciphers. Elliptic curve cryptosystems are often vulnerable to side-channel attacks, and this will be discussed more thoroughly in Chapter IV. For the moment we will ignore these general attacks and focus on those attacks that are specific to ECIES.

III.3.1. Weak Key Generation Algorithm. On one hand it is easy to see how a flawed key generation algorithm can reduce the security of a cryptosystem. It is not difficult to understand how an attacker might be able to break a scheme if, for example, the private key x was always a particular number or had a particular form. However, there is a more subtle reason why key generation is an important part of a cryptosystem, and this has particular relevance when talking about security proofs.

It is very important that the key generation algorithm has the ability to produce lots of different key-pairs and each with equal probability. This may seem a bit contradictory. After all, an attacker will only get to attack the scheme after the key-pair has been chosen, so why should it matter whether the key generation algorithm produces one key or lots and lots?

Well, for a start we have assumed that the attacker knows all the elements of the cryptosystem including the key generation algorithm. If the key generation algorithm only ever produces one key, or a very small number of keys, then it is not hard for the attacker to run the key generation algorithm, find out the private key and break the scheme. The obvious solution to this problem is to keep the key generation algorithm hidden from the attacker, as is often the case with real-life instantiations of cryptosystems.

It turns out that this is not a useful thing to do, but for quite a subtle reason. The reason is that the key generation algorithm is a fixed algorithm that doesn't depend on any secret values, so there are always attackers that *assume* something about the key generation algorithm, such as that it always produces one particular key-pair, and attempt attacks based on this assumption. If the key generation algorithm produces a large enough number of different key-pairs, then these attackers will not have a significant success probability as it is very unlikely that a key-pair that is generated by $\mathcal{G}$ will satisfy the assumption that the attacker makes, but this could be a problem for key generation algorithm with small ranges.

It is possibly best to think about this in terms of a simple example. If sk is some fixed private key for an asymmetric encryption scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, then there is an attacker $\mathcal{A}$ who will try to decrypt a challenge ciphertext using the function $\mathcal{D}(\cdot, sk)$ regardless of what the public key is or whether it knows any information about the key generation algorithm. This means that, even if $\mathcal{G}$ is unknown, $\mathcal{A}$ will still have a high success probability if $\mathcal{G}$ outputs the key-pair (pk, sk) with a significant probability. Since we are forced to consider all possible attackers for the scheme $(\mathcal{G}, \mathcal{E}, \mathcal{D})$, we are forced to conclude that $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is a weak encryption algorithm because an attacker exists that will break the scheme with high probability *even though we do not know what that attacker is*. On the other hand, if $\mathcal{G}$ only ever produces (pk, sk) with an insignificant probability, i.e., $\mathcal{G}$ has the ability to output lots of different keys each with equal probability, then $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is secure because the attacker $\mathcal{A}$ will not work in all but an insignificant fraction of the possible cases.

This demonstrates another weakness of security proofs. Whilst a security proof tells us about the security of a scheme in general, it doesn't tell us about how the scheme works with one particular key-pair. It is entirely possible for a scheme to have a good security proof and yet be completely insecure for some set of keys. The best that a security proof can say is that if one generates a key-pair using the key generation algorithm, then the probability that that scheme will be insecure is insignificant; it doesn't say anything about the suitability of any one specific key-pair that might actually be in use.

III.3.2. Invalid Elliptic Curve Point Attacks. Another assumption that has been made throughout this section is that all the data are valid. In real implementations of elliptic curve cryptosystems, elliptic curve points are usually represented as elements of $\mathbb{F}_q \times \mathbb{F}_q$ or $\mathbb{F}_q$ by using some form of point compression. Whenever an attacker has requested the decryption of a ciphertext (U, c, r) , we have always implicitly assumed that U is a valid point on the elliptic curve E and an element of the subgroup generated by P . If this is not explicitly checked, then an attacker can exploit this to break the scheme.

As an example, consider the following simple attack. The attacker finds a point U on E with small prime order p_1 , chooses any message $m \in \mathcal{M}$ and,

for $0 \leq i < p_1$, computes

$$\begin{aligned} (k_1^{(i)} || k_2^{(i)}) &= KD([i]U, l), \\ c^{(i)} &= \text{ENC}(m, k_1^{(i)}), \\ r^{(i)} &= \text{MAC}(c^{(i)}, k_2^{(i)}). \end{aligned}$$

Let $C^{(i)} = ([i]U, c^{(i)}, r^{(i)})$ and $0 \leq j < p_1$ be such that $j \equiv x \pmod{p_1}$. Since $[j]U = [x]U$, where x is the secret key, $C^{(j)}$ will decrypt to give the message m . It is very unlikely that any of the other ciphertexts $C^{(i)}$ (with $i \neq j$) will decrypt to give m . Therefore, with only p_1 requests to the decryption oracle, the attacker can find out the value of the private key $x \pmod{p_1}$.

If an attacker does this for a series of primes $p_1, p_2, \dots, p_k$ such that $p_1 p_2 \dots p_k \geq q$, then it can recover the whole private key x using the Chinese Remainder Theorem (and it will only need to make $p_1 + p_2 + \dots + p_k$ requests to the decryption oracle to do this).

Of course this attack only works if the elliptic curve contains a number of points with small prime orders. However, other variants of this type of attack exist which involve the attacker changing parts of the public parameters, such as the public key Y , the group generator P or even the elliptic curve E itself. These variants may work even if the original attack does not. Details of methods that can be used to check the validity of elliptic curve points and elliptic curves themselves can be found in Section I.5.

III.3.3. Variable Length Symmetric Keys. Another implicit assumption that is made in the security proofs is that the keys produced by the key derivation function are of a fixed length l . This seems like a trivial point, but it can actually be very important, especially when one combines this with some details about the way in which most key derivation functions work. Most key derivation functions compute $KD(U, l)$ by taking the first l bits of an infinite pseudo-random sequence produced from the input U . This means that, for any $0 < l' < l$, $KD(U, l')$ is the same as the first l' bits of $KD(U, l)$. This is not a problem for ECIES providing that we use symmetric keys of a fixed length l – it becomes a problem, however, if we allow l to vary.

In particular, some early versions of ECIES allowed the symmetric cipher to be a variable length Vernam cipher (see Section III.1.3). In this case encryption is given by:

ALGORITHM III.1: ECIES Encryption (Weak Version)

INPUT: A message m , public key Y and the length of the MAC key l' .

OUTPUT: A ciphertext (U, c, r) .

1. Choose $k \in_R \{1, \dots, q\}$.
2. $U \leftarrow [k]G$.

3. $T \leftarrow [k]Y$.
 4. $(k_1 || k_2) \leftarrow KD(T, |m| + l')$.
 5. **Encrypt the message**, $c = m \oplus k_1$.
 6. **Compute the MAC on the ciphertext**, $r \leftarrow MAC(c, k_2)$.
 7. **Output** (U, c, r) .
-

Decryption is given in the obvious way.

This variant of ECIES was proven to be insecure by Shoup [305]. Suppose $m = m_1 || m_2$ is a message such that $|m_1| > 0$ and $|m_2| = l'$. Suppose further that the ciphertext $C = (U, c, r)$ is an encryption of m and that $c = c_1 || c_2$, where $|c_2| = l'$. It is easy to see that the ciphertext $C' = (U, c', r')$, where

$$c' = c_1 \oplus \Delta \quad \text{and} \quad r' = MAC(c', c_2 \oplus m_2),$$

is a valid encryption of the message $m_1 \oplus \Delta$ for any bit-string Δ of length $|m_1|$. Hence an attacker can check whether a ciphertext C is the encryption of a message m or not by checking to see if the shortened ciphertext C' is an encryption of the shortened message $m_1 \oplus \Delta$ or not. This would allow an IND-CCA2 attacker to break ECIES.

III.3.4. Using the x -Coordinate. As has already been mentioned in Section I.4, in some implementations of ECIES the key derivation function KD is not applied to the elliptic curve point T but to the x -coordinate of T . Whilst this is not really a problem for the cipher, since it is easy to see that computing the x -coordinate of T is roughly as hard as computing the whole of T , it is a problem for the security proof.

The problem occurs because the decryption of a ciphertext (U, c, r) in this case is always the same as the decryption of the ciphertext $(-U, c, r)$. Therefore an attacker with CCA2 access to a decryption oracle can always decrypt the challenge ciphertext $C^* = (U^*, c^*, r^*)$ by requesting the decryption of $(-U^*, c^*, r^*)$. Hence the scheme is not secure against CCA2 attackers. A cryptosystem for which there is some easy way of computing two different ciphertexts that decrypt to give the same message is said to be *benignly malleable* [305].

We can remove this problem by defining a binary relation $\sim$ between ciphertexts, where $C \sim C'$ if (1) $\mathcal{D}(C, sk) = \mathcal{D}(C', sk)$ and (2) it is easy to find C' from C . In this case the decryption oracle must refuse to decrypt not only the challenge ciphertext C^* but any ciphertext C' such that $C^* \sim C'$. These ideas are slowly gaining more acceptance in the academic community [58].

III.4. ECIES-KEM

One problem with hybrid encryption is that one often ends up proving the same results again and again. We can use the three security proofs for ECIES in given Section III.2 as an example: each of these had a different

technique for dealing with the asymmetric (elliptic curve) part of the system, and yet all three used exactly the same arguments about the symmetric part.

Recently Shoup (see [304] and [93]) introduced the concept of a generic hybrid construction (sometimes called a KEM/DEM construction). Here the asymmetric and symmetric parts of a hybrid scheme are separated and each has its own formal security requirements. This allows us to evaluate their security individually and to “mix and match” the asymmetric KEMs and the symmetric DEMs without needing to produce another long, boring security proof.

III.4.1. KEMs and DEMs. A KEM/DEM construction is made of two parts: an asymmetric *key encapsulation mechanism* (or KEM) and a symmetric *data encapsulation mechanism* (or DEM). A KEM takes as input a public key and produces a random symmetric key of a predetermined length and an encryption (or *encapsulation*) of that key. A DEM takes as input a message and a symmetric key and produces an encryption of that message under that key. Once again, it is very important to be precise about how we define a KEM and a DEM.

DEFINITION III.11. *A KEM is a triple of algorithms $(\mathcal{G}, \text{Encap}, \text{Decap})$, where*

- $\mathcal{G}$ is a probabilistic key generation algorithm. It takes no input, except randomness, and outputs a key-pair (pk, sk) . Again pk is the public key and must be distributed to everyone that uses the encryption algorithm, and sk is the private key and should only be made available to those parties who have permission to decrypt messages.
- Encap is a probabilistic algorithm called the encapsulation algorithm. It takes only the public key as input, and randomness, and outputs a symmetric key K of some predetermined length and an encapsulation C_1 of K .
- Decap is a deterministic algorithm called the decapsulation algorithm. It takes as input the private key sk and an encapsulation C_1 . It returns either a symmetric key K or an error symbol $\perp$.

A KEM is *sound* if for all valid key-pairs (pk, sk) and any output $(K, C_1) = \text{Encap}(pk)$ we have $K = \text{Decap}(C_1, sk)$.

It should be fairly easy to see the similarities between KEMs and public key encryption schemes. The definitions are identical except for the fact that the encapsulation algorithm does not take a message as input but rather generates its own (the symmetric key).

DEFINITION III.12. *A DEM is a pair of algorithms (ENC, DEC) , where*

- ENC is a (possibly) probabilistic algorithm called the encryption algorithm. It takes as input a message m and a symmetric key K of some predetermined length and outputs an encryption C_2 of the message m .

- *DEC is a deterministic algorithm called the decryption algorithm. It takes as input an encryption C_2 and a symmetric key K and returns either a message m or an error symbol $\perp$.*

A DEM is sound if for all keys K of the correct length and messages m we have $\text{DEC}(\text{DEC}(m, K), K) = m$.

In order to form a public-key encryption scheme one combines a KEM and a DEM, where the symmetric keys that are outputted by the KEM are the correct size for use by the DEM, in the following manner. Key generation is given by the key generation algorithm for the KEM, $\mathcal{G}$.

ALGORITHM III.2: KEM/DEM Encryption

INPUT: A message m and the public key pk .

OUTPUT: A ciphertext $C = (C_1, C_2)$.

1. $(K, C_1) \leftarrow \text{Encap}(pk)$.
 2. $C_2 \leftarrow \text{Enc}(m, K)$.
 3. Output $C = (C_1, C_2)$.
-

ALGORITHM III.3: KEM/DEM Decryption

INPUT: A ciphertext $C = (C_1, C_2)$ and the private key sk .

OUTPUT: A message m or an “Invalid Ciphertext” message.

1. $K \leftarrow \text{Decap}(C_1, sk)$.
 2. If $K = \perp$ then output “Invalid Ciphertext”.
 3. $m \leftarrow \text{DEC}(C_2, K)$.
 4. If $m = \perp$ then output “Invalid Ciphertext”.
 5. Output m .
-

III.4.2. Security for KEMs and DEMs. The security criteria for KEMs and DEMs are actually quite similar, and both are closely related to the IND game that was used to define the security of a public-key encryption scheme. Both are concerned with the effects of a two-stage attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ who is playing a game against a challenger. For a KEM the attacker is trying to tell a real symmetric key from a random symmetric key and for a DEM the attacker is trying to tell if a particular ciphertext is the encryption of one message or another.

DEFINITION III.13. *For a KEM, the indistinguishability IND game for an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ consists of four major steps:*

1. *A challenger generates a random key-pair (pk, sk) by running the key generation algorithm $\mathcal{G}$.*

2. The attacker runs $\mathcal{A}_1$ on the input pk . It returns some state information s .
3. The challenger generates a valid symmetric key and its encapsulation (K_0, C^*) by running the encapsulation algorithm $\text{Encap}(pk)$. It also generates a random symmetric key K_1 of the same length and chooses a bit $\sigma \in \{0, 1\}$ uniformly at random.
4. The attacker runs $\mathcal{A}_2$ on the input (K_σ, C^*, pk, s) . It returns a guess σ' for σ .

The attacker wins the game if $\sigma' = \sigma$. The advantage of an attacker in playing the IND game is defined to be

$$|Pr[\sigma' = \sigma] - 1/2|.$$

Attackers that are playing the IND game for a KEM can have either CPA, CCA1 or CCA2 access to a decapsulation oracle, i.e., an oracle that runs the decapsulation algorithm for them. For more details see Definition III.4. In order to guarantee that the overall hybrid scheme is secure against IND-CCA2 attackers we will require that the KEM be secure against IND-CCA2 attackers too.

DEFINITION III.14. For a DEM, the indistinguishability IND game for an attacker $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ consists of four major steps:

1. A challenger generates a random symmetric key K of an appropriate size.
2. The attacker runs $\mathcal{A}_1$. It returns two messages m_0 and m_1 of equal size, as well as some state information s .
3. The challenger chooses a bit $\sigma \in \{0, 1\}$ uniformly at random. It computes the challenge ciphertext $C^* = \text{ENC}(m_\sigma, K)$.
4. The attacker runs $\mathcal{A}_2$ on the input (C^*, s) . It returns a guess σ' for σ .

Again, the attacker wins the game if $\sigma' = \sigma$. The advantage of an attacker in playing the IND game is defined to be

$$|Pr[\sigma' = \sigma] - 1/2|.$$

For our purposes, an attacker that is playing the IND game for a DEM can either have no access to a decryption oracle, in which case the attacker is said to be passive (PA), or have complete access to a decryption oracle, in which case the attacker is said to be active (CCA). Obviously an active attacker is forbidden from requesting the decryption of the challenge ciphertext from the decryption oracle. Note that in neither of these two cases can the attacker *encrypt* messages under the randomly generated symmetric key K . This is a major difference between the symmetric and asymmetric cases.

One of the main points of this approach is that the security of the two components can be evaluated independently, as the following theorem of Shoup [304] demonstrates.

THEOREM III.15. *If $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is an asymmetric encryption scheme composed of a KEM that is secure against IND-CCA2 attackers and a DEM that is secure against IND-CCA attackers, then $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ is secure against IND-CCA2 attackers.*

The proof of this theorem is similar to parts of the proofs of security for ECIES that we have already sketched. Suppose there exists an attacker $\mathcal{A}$ that breaks the hybrid scheme in the IND-CCA2 model with a significant advantage. We attempt to build an IND-CCA2 attacker $\mathcal{B}$ that breaks the KEM. Either $\mathcal{B}$ has a significant advantage or $\mathcal{A}$ must be attacking the symmetric part of the scheme alone, and in that case $\mathcal{A}$'s advantage would be the same even if we replaced the symmetric keys used to encrypt the challenge ciphertext with randomly generated symmetric keys. We can therefore use $\mathcal{A}$ to build an IND-CCA attacker $\mathcal{C}$ that attacks the DEM with significant advantage. On the other hand, if both the KEM and the DEM, are secure then the overall hybrid encryption scheme must be secure too.

There is one note of caution however. Granboulan [153] has shown that it is vitally important that the KEM and the DEM are strictly independent of each other, i.e., no information used by the KEM (including any element of the public key or the system parameters) should be made available to the DEM, or Theorem III.15 will not hold.² In other words, if a secure KEM and a secure DEM share some information, then it is possible that the hybrid scheme formed from them might not be secure.

III.4.3. ECIES-KEM and ECIES-DEM. Both ECIES-KEM and ECIES-DEM (both of which are defined in Section I.4) meet their required security levels and have proofs of security to attest to this.

The security proof for ECIES-DEM is the same as the argument used in Section III.2.1 to show that ECIES is secure if the symmetric keys are randomly generated. Essentially one can prove that ECIES-DEM is secure provided the underlying symmetric cipher and MAC scheme are secure. See Section III.1.3 for more details on the security of symmetric primitives.

As one can imagine, the security proof for ECIES-KEM is a little more complicated, and again we are going to have to make some kind of simplifying assumption in the security analysis, just as we had to do for ECIES. It should be easy to see that we can prove the security of ECIES-KEM using either the random oracle model or the generic group model (see Sections III.2.2 and III.2.3). In both models the security proof for ECIES involved demonstrating that either we could break some underlying mathematical problem or the symmetric keys were indistinguishable from completely random keys. This is exactly the condition we need to prove in order to prove the security of ECIES-KEM.

²Similarly, it is important that the DEM is chosen before the asymmetric key pair is generated or else elements of the key-pair may influence the choice of DEM. This would also invalidate Theorem III.15.

We also managed to show this based on the assumption that the hash Diffie–Hellman problem is difficult to solve, but this does not help us prove the security of ECIES-KEM. This is because the hash Diffie–Hellman problem is exactly the same as the problem of breaking ECIES-KEM with an IND-CCA2 attacker. So basing a security proof for ECIES-KEM on the hash Diffie–Hellman problem would be rather like proving that ECIES-KEM is difficult to break whenever it is difficult to break ECIES-KEM!

ECIES-KEM also suffers from all of the problems associated with ECIES discussed in Section III.3, although the use of the cofactor mode (see Section I.4) can help stop the invalid elliptic curve point attacks discussed in Section III.3.2. Nevertheless, ECIES-KEM has already become a popular algorithm with both academic cryptographers and software developers and has already been proposed for inclusion in many important cryptographic standards.

Part 2

Implementation Techniques

CHAPTER IV

Side-Channel Analysis

E. Oswald

Side-channel analysis (SCA) or information leakage analysis (ILA), refers to a new and emerging type of cryptanalysis that uses leaked side-channel information from a cryptographic device to determine the secret key. The traditional cryptographic model consists of two parties, Alice and Bob, who wish to communicate secretly over an insecure channel. To protect against a potential eavesdropper, cryptographic algorithms are used to ensure the secrecy of the communication. In this traditional model, the security of the system relies solely on the secrecy of the key and the mathematical properties of the cryptographic algorithm used.

However, in a more practical scenario the communicating parties are mostly computing devices, such as personal computers, smart cards or mobile phones. On such devices, a cryptographic algorithm is implemented and executed whenever a cryptographic protocol requires its computation. Although these devices operate in potentially hostile environments and are subject to attacks themselves, they are supposed to protect the secret key. Furthermore, such devices can act as a source of information. Depending on the data they process, they might consume different amounts of power, emit different amounts of electromagnetic emanations, produce different types of noises, need different running times or output different error messages; see Figure IV.1. If the device is executing a cryptographic algorithm (which uses a secret key), the leaked information may be directly related to this secret key. In certain cases, this additional information can be exploited to deduce information about the secret key. Hence, the security of the whole system relies on the security of the algorithms and protocols, and on the security of their implementation.

This chapter is as organized as follows. The introductory part starts with a brief section on cryptographic hardware in Section IV.1. Active attacks assume that an attacker can manipulate a device in some way. Such attacks are briefly sketched in Section IV.2. In passive attacks, which are treated in Section IV.3, an attacker monitors the side-channels of a device but does not manipulate the device himself. We introduce the currently used side-channels in Section IV.3.1 and explain the concepts of simple side-channel analysis and differential side-channel analysis in Sections IV.3.2 and IV.3.3. We conclude

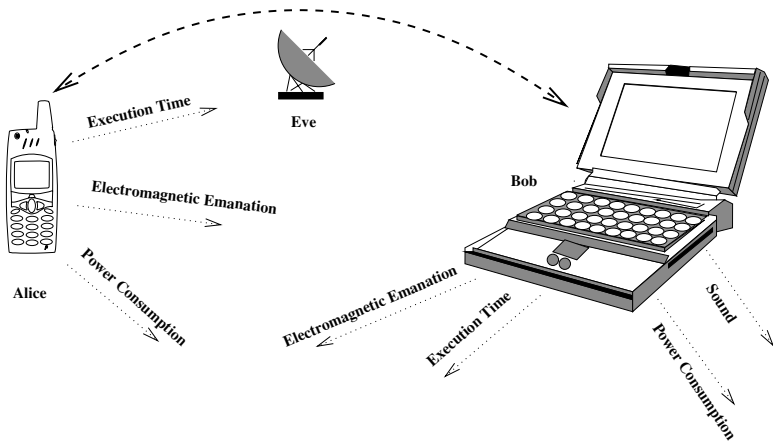


FIGURE IV.1. A practical scenario where Alice and Bob are computing devices and Eve can spy on the communication both directly and via the emanations produced by the devices.

the introductory part of this chapter with a discussion on the attack scenarios for elliptic curve cryptosystems in Section IV.3.4.

The second part is devoted to side-channel analysis on ECC. In Section IV.4 we discuss the application of simple side-channel analysis on implementations of the elliptic curve scalar point multiplication. In Section IV.5 we apply differential side-channel analysis on implementations of the elliptic curve scalar point multiplication.

Defences against side-channel analysis are discussed separately, in Chapter V.

IV.1. Cryptographic Hardware

Traditionally, the main task of cryptographic hardware was the acceleration of operations frequently used in cryptosystems or the acceleration of a complete cryptographic algorithm. Nowadays, hardware devices are also required to store secret keys securely. Especially in applications involving digital signatures, the secrecy of the private signing key is vital to the security of the whole system. A cryptographic device must make key extraction impossible. Active attacks that target cryptographic devices are commonly referred to as *tamper attacks*.

Tamper resistant devices are supposed to make key extraction impossible, while tamper evident devices should make the key extraction obvious. An example of a highly tamper resistant device is IBM's 4758 [IBM CoPro]. It is the only device evaluated against the highest level of tamper resistance (FIPS 140-1[FIPS 140.1], level 4) at the time of writing. Such high-end devices are costly and therefore usually used in few applications such as ATM machines.

Tamper resistant devices that are used for the mass market today are mostly smart cards. Smart cards have to be cheap and are therefore restricted in computing power, memory, and of course, mechanisms to counteract tamper attacks.

IV.1.1. Smart Cards. A smart card usually consists of a microprocessor (8-bit for low cost cards and up to 32-bit in the newest generation of smart cards), memory and some interface. Usually, the interface is a serial interface, for example in the case of banking cards, but also USB-interfaces or RF-interfaces for contactless smart cards are possible. Smart card commands are coded in Application Protocol Data Units (APDUs).

For several purposes, authentication commands such as *internal authenticate* or *external authenticate* are usually available for smart cards with cryptographic functions. The internal authenticate command usually takes some input data and computes some sort of signature or decryption using a secret or private key. The external authenticate command even allows the supply of the key for the cryptographic operation. These two commands are commonly used for attacks in practice.

IV.2. Active Attacks

Active attacks or tamper attacks have a long history in the field of cryptography. The term *active* implies that an attacker actively manipulates the cryptographic device.

IV.2.1. Simple Attacks on Smart Cards. It is relatively simple to extract the microprocessor from a given smart card. Then, the microprocessor can be depackaged and placed into a more suitable medium in order to perform some experiments.

Various simple types of tamper attacks have been popular for smart cards. The very first generations of smart cards used an external programming voltage to read and write contents to its memory. By cutting of this external programming voltage, an attacker could freeze the contents of the memory. Manufacturers' test circuits led to other attacks on smart cards. Such test circuits are used during the testing phase after the fabrication of the smart card. After the testing phase, the test circuits are disconnected from the microprocessor. An attacker has to find and repair the disconnected wires to use the test circuit.

Most of these simpler attacks on smart cards were not only active attacks but were also intrusive. In modern smart cards, the components of the processor are covered by a protective mesh. This mesh reports when it is damaged, at which point the smart card destroys its memory contents or stops functioning.

IV.2.2. Fault Attacks. Here, an attacker tries to induce faulty signals in the power input (or clock input) of the smart card. Often, this forces a change in the data in one of the registers. As a consequence, the computation proceeds with the faulty data. Alternatively, an attacker might try to supply faulty data directly (see also Section III.3.2). Fault attacks on implementations of elliptic curve cryptosystems have been introduced in [26] and were also discussed in [78]. The main idea behind the attacks in this chapter is the following. By inserting a fault in some register, we force the device to execute the point multiplication algorithm with values that are not a point on the given curve but on some other curve instead. This other curve is probably a cryptographically weaker curve that can be used to calculate the private key k more easily. As a consequence, a device has to check its input and its output values for correctness.

IV.3. Passive Attacks

Passive attacks were only recognized in the cryptographic community as a major threat in 1996, when Paul Kocher published the first article about timing attacks [205]. In a passive attack, the attacker only uses the standard functionality of the cryptographic device. In the case of smart cards, these are the internal and the external authenticate commands. The outputs of the device are then used for the attack. A device can have different types of outputs. If such outputs unintentionally deliver information (about the secret key), then the outputs deliver side-channel information and we call them side-channel outputs (or side-channels). In the next sections, we present side-channels that have been exploited in attacks so far.

IV.3.1. Types of Side-Channels. The first official information related to a side-channel analysis dates back to 1956. Peter Wright reported in [349] that MI5, the British intelligence agency, was stuck in their efforts to break an encryption machine in the Egyptian Embassy in London. The encryption machine was a rotor machine of the Hagelin type. Breaking the machine's encryption exceeded their computational capabilities. Wright suggested placing a microphone near that machine to spy on the click-sound the machine produced. Wright discovered that the sound of the clicks allows the attacker to determine the position of some of the rotors. This additional information reduced the computation effort to break the cipher, and MI5 was able to spy on the communication for years.

Timing Leakage : Timing information about the execution of a cryptographic algorithm was the first side-channel that was brought to the attention of the cryptographic community by Kocher. In [205] he introduced a technique to exploit even very small timing variations in an attack. Timing variations often occur because of data-dependent instructions. For example,

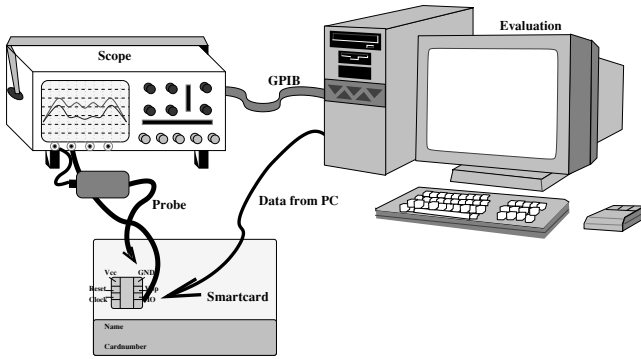


FIGURE IV.2. This is a typical measurement setup to conduct power (or EM) measurements. A personal computer is used to operate the smart card and is connected with the digital scope. The scope measures the power consumption on the GND contact with a probe. A second probe, which is connected to the IO-contact, is used as a trigger. The measurements are sent from the scope to the personal computer for evaluation.

in the (original) Montgomery multiplication [254], a final subtraction step is needed only if the result is exceeding the modulus.

In order to exploit timing information, very precise timing measurements need to be made.

Power Consumption Leakage : Nowadays, almost all smart card processors are implemented in CMOS (Complementary Metal-Oxide Silicon) technology. The dominating factor for the power consumption of a CMOS gate is the dynamic power consumption [348]. Two types of power consumption leakage can be observed. The *transition count leakage* is related to the number of bits that change their state at a time. The *Hamming weight leakage* is related to the number of 1-bits, being processed at a time. The internal current flow of a smart card can be observed on the outside of the smart card by, for example, putting a small resistor between the ground (GND) of the smart card and the true ground. The current flowing through the resistor creates a voltage that can be measured by a digital oscilloscope (see Figure IV.2).

The first practical realization of an attack that makes use of the power consumption leakage was reported by Kocher et al. in [206]. Since then, many companies and universities have developed the skills to conduct these measurements in practice.

Electromagnetic Radiation Leakage : The movement of electrically charged particles causes electromagnetic leakage. In CMOS technology, current flows whenever the circuit is clocked and gates switch. This current flow is visible in the power consumption, and it is also visible as electromagnetic emanations. There are two main categories for electromagnetic emanations. *Direct emanations* result from intentional current flow. *Unintentional emanations* are caused by the miniaturization and complexity of modern CMOS devices.

The first concrete measurements were presented in [139] and [278]. At the time of writing, it is possible to conduct EM attacks with measurements taken several meters away from the attacked device [5].

Combining Side-Channels : The combination of several different side-channels has not been investigated intensively so far. Two sources of side-channel information that are known to be efficiently combined are the timing and the power side-channel. For instance, power measurements can be used to get precise timing measurements for intermediate operations (if these operations are clearly visible in the power trace). These power measurements can then be used in timing attacks. This observation has been used in [342] and [289].

Error Messages : Error-message attacks usually target devices implementing a decryption scheme. It is a common assumption that a device gives a feedback about whether or not a message could be decrypted successfully. If an attacker knows the reason why a decryption failed, information about the secret key can be deduced by sending well-chosen ciphertexts to the device.

These kind of attacks were first introduced by Bleichenbacher in [30]. This article describes a chosen ciphertext attack against the RSA encryption standard PKCS#1. Other error-message attacks are mentioned in [232] against RSA-OAEP, [99] and [189] against the NESSIE candidate EPOC-2 and [330] against the padding and subsequent encryption in CBC mode as it is done in various standards.

IV.3.2. Simple Side-Channel Analysis. In simple side-channel analysis, an attacker uses the side-channel information from *one* measurement directly to determine (parts of) the secret key. Therefore, the side-channel information related to the attacked instructions (the signal) needs to be larger than the side-channel information of the unrelated instructions (the noise). In order to be able to exploit this information, at least some of the executed instructions need to be distinguishable by their side-channel trace. In addition, the attacked instructions need to have a relatively simple relationship with the secret key.

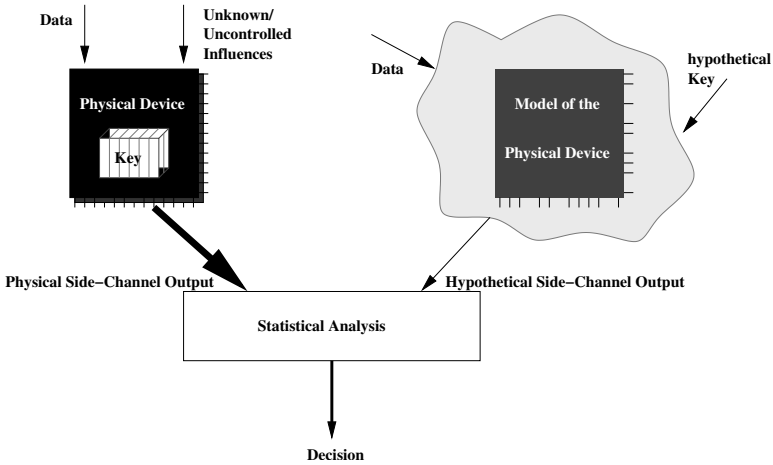


FIGURE IV.3. The idea behind differential side-channel analysis: an attacker compares the predictions from his hypothetical model with the side-channel outputs of the real, physical device.

The question, of whether the signal is clearly visible in the side-channel trace, often depends on the underlying hardware, but also sometimes on the implementation itself; see Chapter V.

IV.3.3. Differential Side-Channel Analysis. When simple side-channel analysis are not feasible due to too much noise in the measurements, differential side-channel analysis can be tried. Differential side-channel analysis uses many measurements to filter out the interfering noise. Simple side-channel analysis exploits the relationship between the executed instructions and the side-channel output. Differential side-channel analysis exploits the relationship between the processed data and the side-channel output.

In differential side-channel analysis, an attacker uses a so-called hypothetical model of the attacked device. The quality of this model is dependent on the capabilities of the attacker:

Clever Outsiders: They do not have insider knowledge of the system. They are assumed to have moderately sophisticated equipment.

Knowledgeable Insiders: They are educated in the subject and have a varying degree of understanding of the attacked system. They have sophisticated tools to conduct the measurements.

Funded Organizations: They have teams of specialists that may have access to insider knowledge about the attacked system, and they have the most advanced analysis tools.

The model is used to predict the side-channel output of a device. Used as such a predictor, it can output several different values. These can be values

predicting the leakage of one side-channel for several moments in time, but they can also be values predicting the leakage of different side-channels. In case only one single output value is used for an attack, then the attack is called a *first-order attack*. If two or more output values for the same side-channel are used in an attack, then the attack is called a *second-order attack* or *higher-order attack*, respectively.

These predictions are compared to the real, measured side-channel output of the device. Comparisons are performed by applying statistical methods on the data. Among others, the most popular methods are the *distance-of-mean test* and *correlation analysis*.

For the distance-of-mean test, the model only needs to predict two values, low and high, for a certain moment of time in the execution. These two predictions are used to classify the measurement data. All measurements for which the prediction stays high are put into one set, and all other measurements are put in a second set. Then, the difference between the sets is calculated by computing the difference of their mean values.

For a formal description of this test, we use the following notation. Let t_i denote the i th measurement data (i.e., the i th trace) and let t_i^l and t_i^h denote a trace for which the model predicted a *low* and a *high* side-channel leakage, respectively. A set of traces t_i^l , t_i^h and t_i is then denoted by T^l , T^h and T and the average trace of such a set by $\bar{T}^l$, $\bar{T}^h$ and $\bar{T}$. Consequently, the difference trace for the distance-of-mean method is calculated by

$$D = \bar{T}^l - \bar{T}^h.$$

Whenever this difference is sufficiently large, or has some other recognizable properties, it can be used to verify or reject the key hypothesis.

For correlation analysis, the model predicts the amount of side-channel leakage for a certain moment of time in the execution. These predictions are correlated to the real side-channel output. To measure this correlation, the Pearson correlation coefficient can be used. Let p_i denote the prediction of the model for the i th trace and P set of such predictions. Then the Pearson correlation coefficient is given as

$$C(T, P) = \frac{E(T * P) - E(T) * E(P)}{\sqrt{\text{Var}(T) * \text{Var}(P)}}, \quad -1 \leq C(T, P) \leq 1.$$

Here $E(T)$ denotes the expectation (average) trace of the set of traces T and $\text{Var}(T)$ denotes the variance of a set of traces T . If this correlation is high, i.e., close to +1 or -1, then the key hypothesis is assumed to be correct.

To summarize and conclude this section, differential side-channel analysis is possible if an attacker can make a hypothesis over a small part of the secret key to predict the side-channel output with his model.

IV.3.4. Target Operations in Elliptic Curve Cryptosystems. There are two operations in commonly used elliptic curve cryptosystems that involve

the private key or an ephemeral (secret) key. We briefly discuss the scenarios under which these operations are attacked.

- The modular multiplication of a known value and the private key needs to be computed in ECDSA. If the multiplication is implemented in such a way that the multiplier is the private key and the multiplication is carried out with a variant of the binary algorithm, then this implementation is, in principle, vulnerable to side-channel analysis. However, this operation can be implemented securely by not using the private key as a multiplier.
- The scalar multiplication of a secret value with a known elliptic curve point, i.e., the *point multiplication* [ECC, Section IV.2], needs to be calculated in all elliptic curve cryptosystems. Also this operation is usually implemented by some version of the binary algorithm. It has been shown to be vulnerable to simple and differential side-channel analysis.

Probably the most important elliptic curve protocol is ECDSA. From the previous considerations it is clear that the operation that is difficult to secure is the point multiplication operation. It is also well known [262] that not all bits of the ephemeral ECDSA key need to be known in order to reconstruct the private key of the ECDSA. As a result, the point multiplication operation must be implemented to resist in particular simple side-channel analysis.

IV.4. Simple SCA Attacks on Point Multiplications

In the case of a simple SCA attack on an implementation of a point multiplication algorithm, the adversary is assumed to be able to monitor the side-channel leakage of *one* point multiplication, $Q = [k]P$, where Q and P are points on an elliptic curve E and $k \in \mathbb{Z}$ is a scalar. The attacker's goal is to learn the key k using the information obtained from carefully observing the side-channel leakage (e.g. power trace) of a point multiplication. Such a point multiplication consists of a sequence of point addition, point subtraction and point doubling operations. Each elliptic curve operation itself consists of a sequence of field operations. In most implementations, the standard sequence of field operations in point addition differs from that in point doubling. Every field operation has its unique side-channel trace. Hence, the sequence of field operations of point addition has a different side-channel pattern than that of point doubling (see Figures IV.4 and IV.5 for an example with power traces).

IV.4.1. Attacking the Basic Double-and-Add Algorithm. It was already observed in [87] that the implementation of the simplest form of a double-and-add algorithm, which is the binary algorithm (see Algorithm IV.1), is an easy target for simple side-channel analysis.

First we note that the conditional branch (i.e., Step 5 in Algorithm IV.1) only depends on a single bit of k . Hence, an attacker can inspect the power

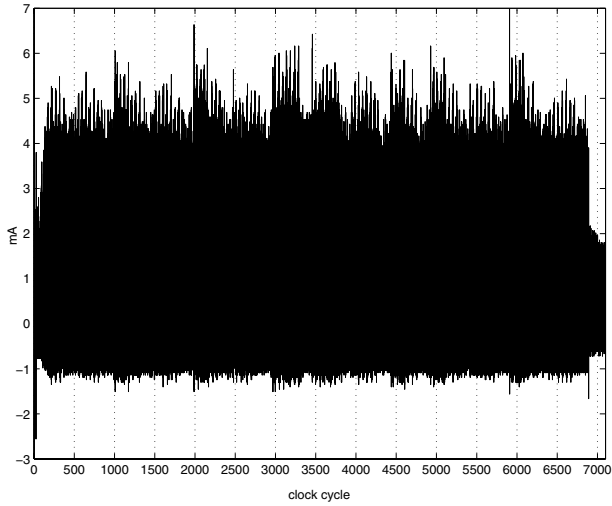


FIGURE IV.4. A point addition power trace.

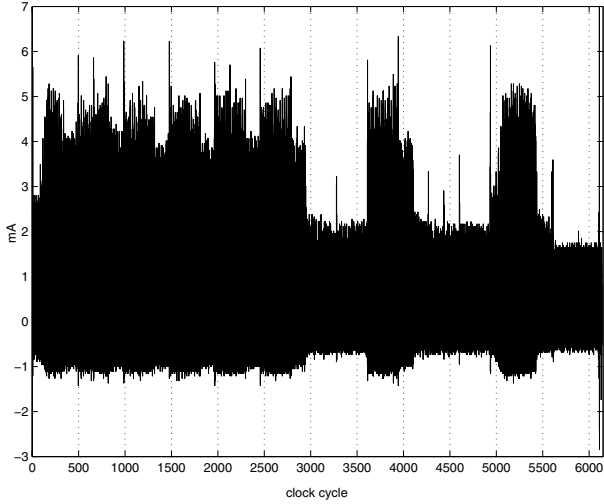


FIGURE IV.5. A point doubling power trace.

trace of the point multiplication algorithm to determine k . We have assumed that it is possible to distinguish the point addition operation from the point doubling operation in the power trace. Since a point addition operation can

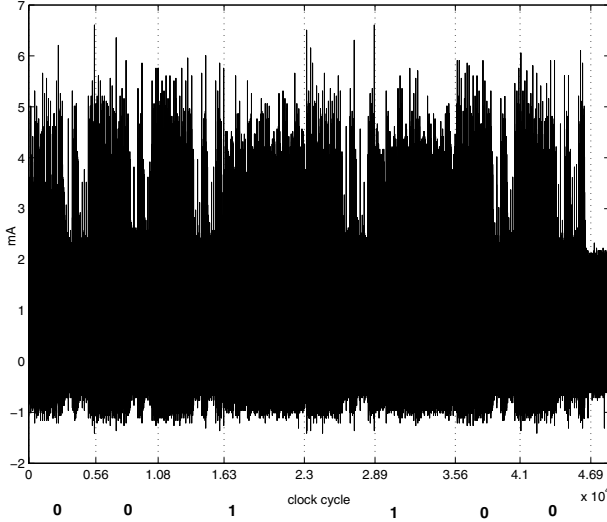


FIGURE IV.6. The power consumption trace of an elliptic curve scalar point multiplication operation that was performed with a simple double-and-add algorithm. The used scalar can be read directly from this power consumption trace.

only be caused by a 1 in the bit representation of k , deducing the key is trivial (see Figure IV.6 for an example) in this naive implementation.

ALGORITHM IV.1: Right-to-Left Binary Algorithm

INPUT: Point P and ℓ -bit multiplier $k = \sum_{j=0}^{\ell-1} k_j 2^j$, $k_j \in \{0, 1\}$.

OUTPUT: $Q = [k]P$.

1. $Q \leftarrow P$.
 2. If $k_0 = 1$ then $R \leftarrow P$ else $R \leftarrow \mathcal{O}$.
 3. For $i = 1$ to $\ell - 1$ do:
 4. $Q \leftarrow [2]Q$.
 5. If $(k_i = 1)$ then $R \leftarrow R + Q$.
 6. Return R .
-

More interesting to attack are algorithms that make use of the rich arithmetic on elliptic curves. This arithmetic allows the use of other representations of k , instead of only the binary representation. They offer some inherent resistance to straightforward simple SCA because they also make use of point subtraction. Because point addition and point subtraction only differ slightly, they can be implemented in such a way that their side-channel patterns look

alike. Therefore, an attacker cannot determine the key bits by just inspecting the side-channel trace.

IV.4.2. Attacking Double-Add-and-Subtract Algorithms. The task of an attacker is to deduce some information about k by using the information of the side-channel trace. In particular, an attacker wants to determine and exploit the relationship between the occurrence of certain sequences of bits and certain sequences of side-channel patterns. Let X be a random variable that denotes the sequence of patterns in the side-channel trace, i.e., an *AD*-sequence (for example $X = \text{“DDD”}$ or $X = \text{“DAD”}$). Let Y be a random variable that denotes a sequence of patterns in the digit representation of k , i.e., a 01-sequence (for example $Y = 000$ or $Y = 01$). Then an attacker is interested in exploiting and calculating the conditional probability

$$\Pr(Y = y|X = x) = \frac{\Pr(Y = y \cap X = x)}{\Pr(X = x)} \quad (\text{IV.1})$$

for many different x and y . Such probabilities can be calculated by using Markov theory [155].

A Markov process, or in case of a finite state space also called Markov chain, is a rather simple statistical process. In a Markov process, the next state is dependent on the present state but independent of the way in which the present state arose from the states before. The transitions between the states are determined by random variables that are either known or have to be estimated.

DEFINITION IV.1. Let T be a $(k \times k)$ matrix with elements $t_{ij}, 1 \leq i, j \leq k$. A random process $(X_0, X_1, \dots)$ with finite state space $S = \{s_1, \dots, s_k\}$ is said to be a **Markov chain** with transition matrix T if, for all n all $i, j \in \{1, \dots, k\}$, and all $i_0, \dots, i_{n-1} \in \{1, \dots, k\}$ we have

$$\begin{aligned} \Pr(X_{n+1} = s_j | X_0 = s_{i_0}, \dots, X_n = s_i) = \\ \Pr(X_{n+1} = s_j | X_n = s_i) = t_{ij}. \end{aligned}$$

A large class of such Markov processes has the two important properties of being irreducible and aperiodic.

The first property, i.e., that a process is *irreducible*, means that all states can be reached from all other states with a finite number of steps. The second property, i.e., that a process is *aperiodic*, means that all states are aperiodic. A state is aperiodic if the period is equal to 1, i.e., the probability of returning to a state is always positive. These two properties are conditions for the main theorem of Markov theory. Before we state this theorem we define the stationary distribution first.

DEFINITION IV.2. Let $(X_0, X_1, \dots)$ be a Markov chain with state space given by $\{s_1, \dots, s_k\}$ and transition matrix T . A row vector $\pi = (\pi_1, \dots, \pi_k)$ is said

to be a **stationary distribution** for the Markov chain if it satisfies

$$\pi_i \geq 0 \forall i, \text{ and } \sum_{i=1}^n \pi_i = 1, \text{ and} \quad (\text{IV.2})$$

$$\pi T = \pi. \quad (\text{IV.3})$$

THEOREM IV.3. *For any irreducible and aperiodic Markov chain, there exists a unique stationary distribution π , and any distribution μ^n of the chain at time n approaches π as $n \rightarrow \infty$, regardless of the initial distribution μ^0 .*

This theorem states that, for Markov processes having the properties of being aperiodic and irreducible, a *steady state* always exists. By using the transition matrix T and the steady-state vector, we can calculate the conditional probabilities (IV.1).

Example: Consider a simple double-add-and-subtract algorithm as shown in Algorithm IV.2.

ALGORITHM IV.2: Double-Add-and-Subtract Algorithm

INPUT: Point P and ℓ -bit multiplier $k = \sum_{j=0}^{\ell-1} k_j 2^j$, $k_j \in \{0, 1\}$.

OUTPUT: $Q = [k]P$.

1. $R_0 \leftarrow \mathcal{O}, R_1 \leftarrow P, s \leftarrow 0$.
 2. For $i = 0$ to $\ell - 1$ do:
 3. If $k_i = 0$ then
 4. If $s = 11$ then $R_0 \leftarrow R_0 + R_1$.
 5. $s \leftarrow 0, R_1 \leftarrow [2]R_1$.
 6. If $k_i = 1$ then
 7. If $s = 0$ then $R_0 \leftarrow R_0 + R_1, R_1 \leftarrow [2]R_1, s \leftarrow 1$.
 8. If $s = 1$ then $R_0 \leftarrow R_0 - R_1, R_1 \leftarrow [2]R_1, s \leftarrow 11$.
 9. If $s = 11$ then $R_1 \leftarrow [2]R_1$.
 10. If $s = 11$ then $R_0 \leftarrow R_0 + R_1$.
 11. Return R_0
-

There are three different states s in this algorithm. The initial state is always 0. Under the assumption that $\Pr(k_i = 0) = \Pr(k_i = 1) = 1/2$, the transition matrix for this algorithm is

$$T = \begin{pmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 \end{pmatrix}. \quad (\text{IV.4})$$

From the transition matrix the steady-state vector, which is $(1/2, 1/4, 1/4)$, can be calculated. The number of elliptic curve operations that are induced by an ℓ -bit number can be calculated as well. In each state, a point doubling has to be calculated. Hence, there are ℓ point doublings. In addition, in half

of the cases in each state, a point addition has to be calculated. Hence, there are $(1/4 + 1/8 + 1/8)\ell$ point additions. In total, $3/2\ell$ elliptic curve operations are calculated.

A hidden Markov process is a Markov process in which we can only observe a sequence of emissions (the *AD*-sequence), but we do not know the sequence of states, which are related to the key bits (the 01-sequence) the process went through. A hidden Markov model can be characterized by the quintuple (S, O, T, E, s_0) . The notation used here is similar to the one used before: the transition matrix is denoted by T and the finite set of states is called S . The emissions are denoted by the set O . The emission matrix E is a $|S| \times |O|$ matrix and contains the conditional probability that an emission symbol of the set E was produced in state S : $E_{ij} = \Pr(O_j|S_i)$. The initial state distribution is denoted by s_0 . Given a (hidden) Markov model, the task of the attacker is to find for a given *AD*-sequence, a corresponding state sequence that explains the given *AD*-sequence best. One approach to tackle this problem is to choose sub-sequences that are individually most likely, i.e., which have the highest conditional probabilities [269].

Example: Suppose we use Algorithm IV.2 to perform a point multiplication with the scalar $k = 560623$. As a first step, we calculate sufficiently many conditional probabilities (see Table IV.1) by using the Markov model that we derived in the previous example.

TABLE IV.1. Non-zero conditional probabilities. In this table we use an abbreviated notation, i.e., we write $p(000|DDD)$ instead of $p(Y = 000|X = DDD)$. We use the LSB first representation.

$\Pr(000 DDD) = 1/2$	$\Pr(01 DAD) = 1/2$	$\Pr(11 ADAD) = 1/2$
$\Pr(100 DDD) = 1/4$	$\Pr(10 DAD) = 1/4$	$\Pr(10 ADAD) = 1/4$
$\Pr(111 DDD) = 1/4$	$\Pr(11 DAD) = 1/4$	$\Pr(01 ADAD) = 1/4$
$\Pr(001 DDAD) = 1/2$	$\Pr(000 ADDD) = 1/4$	$\Pr(110 ADADAD) = 1/2$
$\Pr(101 DDAD) = 1/4$	$\Pr(100 ADDD) = 1/2$	$\Pr(101 ADADAD) = 1/4$
$\Pr(110 DDAD) = 1/4$	$\Pr(111 ADDD) = 1/4$	$\Pr(011 ADADAD) = 1/4$

Table IV.2 shows how the remainder of the attack works. The first row contains the *AD*-sequence that we deduced from the power trace. In the second row, this sequence is split into sub-sequences. In a practical attack, the length of the sub-sequences will depend on the computational capabilities of the attacker. However, it will be much larger than in this toy example. In the third, fourth and fifth rows, the possible bit-patterns are listed according to their conditional probabilities.

TABLE IV.2. $k = 11110111101100010001$, LSB First Representation

ADADDDADADADDDADADADADDDADDDDDAD							
ADAD	DDAD	ADAD	DDAD	ADAD	ADDD	ADDD	DAD
11	001	11	001	11	100	100	01
10	101	10	101	10	000	000	10
01	110	01	110	01	111	111	11

We are interested in the number of possible scalar values that have to be tested before the correct k is found. Let ℓ denote the number of bits of k and n the average length of the sub-sequences. In the worst case, if we only take the non-zero conditional probabilities into account, but not their individual values, we have to test $0.5 \times 3^{\ell/2n}$ keys on average. Let m be the number of sub-sequences, i.e., $m = 3\ell/2n$ in this example, and $g(x, y) = \binom{x}{y} 2^{x-y}$. If we take the individual conditional probabilities into account, this reduces to $0.5^m + 0.5 \sum_{i=0}^{m-1} 0.5^i 0.25^{m-i} \left(1 + g(m, i) + 2 \sum_{j=i+1}^m g(m, j) \right) g(m, i)$ keys on average [270].

The approach that we used in the previous example was to choose sub-sequences that are individually most likely. This strategy guarantees us finding the key k ; however, it does not take sequences of sub-sequences into account. Another widely used strategy is to find the single best path. A well known technique for this approach is the Viterbi algorithm and has been used successfully to attack double-add-and-subtract algorithms [195]. In case of a randomized version of the algorithm that was discussed in the previous example, the Viterbi algorithm determines approximately 88% of all intermediate states correctly. The remaining and unknown states can be determined by a meet-in-the-middle technique that is sketched in the following section.

Improvements : Assume that the point Q is given by $Q = [k]P$, where $k = xb + y$ denotes the private key k and P the public base-point. Then we know that for the correct values of x , b and y the equation $Q - [y]P = [xb]P$ must hold. By computing and comparing the values of the left- and the right-hand sides of this equation, we can determine k . Hence, if k is almost determined, the remaining bits can be guessed and verified by checking the previous equation. This approximately halves the search space for k .

Other Attack Scenarios : Suppose an attacker has the ability to force a device to use the same scalar k for several point multiplication operations. Due to the randomization, the same scalar will produce different power traces that all give information about the used scalar. Combining the information gained from several traces, a scalar value can usually be determined with very few (approximately 10) measurements [265], [340].

IV.4.3. The Doubling Attack. This attack relies on the fact that similar intermediate values are manipulated when working with a point P and its double $[2]P$ [122]. We assume that an attacker can identify point doubling operations with identical data in two power traces.

ALGORITHM IV.3: Double-and-Add-Always Algorithm

INPUT: Point P and ℓ -bit multiplier $k = \sum_{j=0}^{\ell-1} k_j 2^j$, $k_j \in \{0, 1\}$.
 OUTPUT: $Q = [k]P$.

1. $R_0 \leftarrow P$.
 2. For $j = \ell - 2$ to 0 by -1 do:
 3. $R_0 \leftarrow [2]R_0$.
 4. $R_{1-k_j} \leftarrow R_{1-k_j} + P$.
 5. Return R_0 .
-

Algorithm IV.3 is secure against simple SCA on a first glance as in each iteration a point doubling and a point addition operation is executed. The value of the variable R_0 after $j+1$ iterations is $Q_j(P) = [\sum_{i=0}^{j-1} 2^{j-i}]P$. Rewriting this expression in terms of $[2]P$ leads to $Q_j(P) = Q_{j-1}([2]P) + [d_{\ell-j}]P$. Thus, the intermediate result of the algorithm with input P (which is stored in R_0) at step j is equal to the intermediate result of the algorithm with input $[2]P$ at step $j-1$ if and only if $d_{\ell-j}$ is zero. Hence, we just need to compare the doubling operation at step $j+1$ for P and at step j for $[2]P$ to recover the bit $d_{\ell-j}$.

IV.5. Differential SCA Attacks on Point Multiplications

The first article related to side-channel analysis on public-key cryptosystems was Messerges et al. [245]. Shortly afterwards, Coron published an article specifically related towards implementations of elliptic curve cryptosystems [87]. Goubin published an article wherein some of Coron's proposed countermeasures were counteracted by a refined attack [151], which was analysed further in [313].

IV.5.1. Differential SCA Without any Knowledge of the Implementation. Two differential SCA attacks have been published which assume no knowledge about the attacked device.

The SEMD (Single-Exponent Multiple-Data) attack assumes that an attacker can perform the point multiplication operation with the secret value k and some public value r with several elliptic curve points. Then, for both the secret and the public values, the conducted measurements, tk_i and tr_i , are averaged. The two average traces are subtracted from each other to produce a difference trace. Whenever this difference trace shows a peak, the two values k and r must have been different as well:

$$D = \bar{T}k - \bar{T}r.$$

Thus, by comparing the power traces from a known and an unknown scalar an attacker can learn the unknown scalar.

The MESD (Multiple-Exponent Single-Data) attack works similarly. In this scenario it is assumed that the attacker can feed the device with multiple scalar values and one single elliptic curve point. Another assumption is that the attacker is allowed to perform the point multiplication with the secret scalar value and the same elliptic curve point several times. The measurements for the secret scalar, ts_i , are averaged to get rid of noise. Then, the attacker determines the bits of the secret value step by step. For each choice of a new bit, $r0$ or $r1$, the attacker calculates how well the choice fits the secret exponent:

$$\begin{aligned} D1 &= \bar{T}s - tr1 \\ D0 &= \bar{T}s - tr0. \end{aligned}$$

Whichever difference trace is zero for a longer time corresponds to the correct bit.

IV.5.2. Differential SCA with Knowledge of the Implementation.

The two other published attacks work according to our general introduction in Section IV.3.3. The ZEMD (Zero-Exponent Multiple-Data) uses as a statistical method the distance-of-mean test. The outputs of the model are therefore a simple classification of the measurement data into two sets. For one set, the model predicts a high side-channel leakage for a certain moment in time, T^h , while for the other set a low side-channel leakage T^l is predicted. Only if the model predicted the side-channel behaviour correctly, is the classification into the two different sets meaningful and thus a difference between the sets must be visible:

$$D = \bar{T}^h - \bar{T}^l.$$

Coron's attack is essentially the same as the ZEMD attack, but it uses the correlation coefficient instead of the distance-of-mean test. Both attacks assume that an attacker knows the internal representation of the elliptic curve points and that a prediction of the side-channel output is possible.

IV.5.3. Differential SCA Using Address Information.

As explained in Section IV.3.3, an attacker tries to include as much information as possible in her hypothetical model. This information is usually related to the intermediate data, but it can also be related to the intermediate addresses. In some devices, loading data from a specific address, i.e., performing $data = R[Address]$, is composed into two stages. First, the physical address is determined given $R[Address]$ and then the data that is stored in this address are loaded. As addresses also travel over a bus (as data travel over a bus), different addresses can be expected to leak different information. In [175], this assumption has been verified for a certain device and an attack based on it is presented. This attack breaks implementations of EC schemes on

that device, which resist typical differential SCA attacks based on predicting intermediate data by using a Montgomery point multiplication algorithm (see Algorithm V.2). In Step 4 of this algorithm, data is fetched from a particular address that depends on a particular bit of the key. Hence, when for example performing an SEMD attack, the attacker should see a peak in the side-channel trace whenever two key bits differ.

IV.5.4. Refined Differential SCA. If projective coordinates are used in an implementation, they can be used to randomize the intermediate data; differential SCA as described in the previous sections cannot be used directly. However, under the assumption that the multiplier is not randomized, exploiting the properties of so-called *special points* can lead to an attack; see [151]. A special point $P_0 \neq \mathcal{O}$ is a point having the property that one of the affine or projective coordinates is 0. Hence, randomization does not affect this property.

Suppose the attacker already knows the highest bits $d_{\ell-1}, \dots, d_{j+1}$ of k . To attack the next bit d_j in Algorithm IV.3, the attacker needs to feed suitable multiples P_1 of P_0 into the algorithm. Such a multiple is calculated based on the (guessed) key bit d_j . If this guess is correct, then this multiple is nicely related to the content of the registers in the $(j+1)$ th step: If we denote the value of R_1 in the $(j+1)$ th step by $[x]P$, then, if d_j is correct, the special point is $[x^{-1}]P$ (we have to assume that x is relatively prime to the order of P). Hence, in the $(j+1)$ th step of the algorithm with input P_1 , the point $[xx^{-1}]P_0 = P_0$ is processed. The special property of this point can then be picked up from several side-channel traces by averaging over the traces.

In [6], an extension to [151] is described. This extended attack is based on the observation that, even if a point does not have a zero coordinate, some of the intermediate values that occur during point addition (or point doubling) might become zero.

CHAPTER V

Defences Against Side-Channel Analysis

M. Joye

V.1. Introduction

This chapter is aimed at studying the resistance of elliptic curve cryptosystems against side-channel analysis. The main operation for elliptic curve cryptosystems is the point multiplication: $Q = [k]P$, where the multiplier k is generally a secret (or private) parameter. In particular, this chapter surveys various *software* strategies to prevent side-channel attacks in the computation of $Q = [k]P$.

Given the rich mathematical structure of elliptic curves, an endless number of new countermeasures can be imagined. More importantly, as we will see, efficient countermeasures (i.e., with almost no impact on the running time and/or the memory requirements) are known. The advantage is clearly on the end-user side (not on the attacker side). We must also keep in mind that present-day cryptographic devices come equipped with a variety of hardware countermeasures, already making harder (or impossible) side-channel attacks.

As given in textbooks, the formulæ for doubling a point or for adding two (distinct) points on a Weierstraß elliptic curve are different. So, for example, from the distinction between the two operations, a *simple power analysis* (i.e., a simple side-channel analysis using power consumption as a side-channel; cf. Section IV.4) might produce different power traces, revealing the value of k in the point multiplication algorithm. There are basically three approaches to circumvent the leakage. This can be achieved by:

1. unifying the addition formulæ [51, 50] or considering alternative parameterizations [188, 220, 27];
2. inserting dummy instructions or operations [87, 76];
3. using algorithms that already behave “regularly” [225, 264, 253, 51, 178, 120].

Even if a point multiplication algorithm is protected against simple side-channel analysis, it may succumb to the more sophisticated *differential side-channel analysis* [87, 206] (cf. Section IV.5). Practically, we note that, except ECIES, very few elliptic curve cryptosystems are susceptible to such attacks as, usually, the input point is imposed by the system and the multiplier is an ephemeral parameter, varying at each execution.

In order to thwart differential side-channel analysis, the inputs of the point multiplication algorithm, namely, the base-point P and the multiplier k , should be randomized [87, 190].

The insertion of random decisions during the execution of the point multiplication algorithm also helps in preventing side-channel analysis [271, 156]. Of course, several countermeasures can be combined for providing a higher security level.

V.2. Indistinguishable Point Addition Formulæ

We start by recalling the explicit group law on elliptic curves (see [ECC, Section III.2]). Let E denote an elliptic curve given by the Weierstraß form

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (\text{V.1})$$

and let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ denote points on the curve. The inverse of P_1 is $-P_1 = (x_1, -y_1 - a_1x_1 - a_3)$. If $P_1 \neq -P_2$, then the sum $P_3 = P_1 + P_2 = (x_3, y_3)$ is defined as

$$x_3 = \lambda^2 + a_1\lambda - a_2 - x_1 - x_2, \quad y_3 = \lambda(x_1 - x_3) - y_1 - a_1x_3 - a_3 \quad (\text{V.2})$$

with

$$\lambda = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2} & \text{when } x_1 \neq x_2, \\ \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3} & \text{when } x_1 = x_2. \end{cases} \quad (\text{V.3})$$

From Eq. (V.3), it clearly appears that the formulæ for adding two (distinct) points and for doubling a point are different.

This section discusses various ways for making indistinguishable the addition formulæ on elliptic curves so that a side-channel analysis cannot reveal what kind of addition is being processed.

V.2.1. Unified Addition Formulæ. In [51], Brier and Joye observed that the expressions of slope λ in Eq. (V.3) can be unified for adding *or* doubling points. When $P_1 \neq \pm P_2$ and provided that P_1 and $-P_2$ do not have the same y -coordinate (i.e., provided that $y_1 \neq -y_2 - a_1x_2 - a_3$), we can write $\lambda = \frac{y_1 - y_2}{x_1 - x_2} = \frac{y_1 - y_2}{x_1 - x_2} \cdot \frac{y_1 + y_2 + a_1x_2 + a_3}{y_1 + y_2 + a_1x_2 + a_3}$ and so obtain after a little algebra

$$\lambda = \frac{x_1^2 + x_1x_2 + x_2^2 + a_2x_1 + a_2x_2 + a_4 - a_1y_1}{y_1 + y_2 + a_1x_2 + a_3}. \quad (\text{V.4})$$

Remarkably, this expression for λ remains valid when $P_1 = P_2$ and can thus be used for doubling a point. Indeed, replacing (x_2, y_2) by (x_1, y_1) in Eq. (V.4), we get $\lambda = \frac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}$.

To include the uncovered case $y(P_1) = y(-P_2)$, we can extend the previous expression to

$$\lambda = \frac{(x_1^2 + x_1x_2 + x_2^2 + a_2x_1 + a_2x_2 + a_4 - a_1y_1) + \epsilon(y_1 - y_2)}{(y_1 + y_2 + a_1x_2 + a_3) + \epsilon(x_1 - x_2)}$$

for any $\epsilon \in \mathbb{K}$ [50]. For example, when $\text{char}(\mathbb{K}) \neq 2$, we can set $\epsilon = 1$ if $y_1 + y_2 + a_1x_2 + a_3 + x_1 - x_2 \neq 0$ and $\epsilon = -1$ otherwise; when $\text{char}(\mathbb{K}) = 2$, we can respectively set $\epsilon = 1$ and $\epsilon = 0$. Another strategy consists in randomly choosing $\epsilon \in \mathbb{K}$. The extended expressions for λ prevent the exceptional procedure attacks of [179].

Two main classes of elliptic curves are used in cryptography: elliptic curves over large prime fields and elliptic curves over binary fields, i.e., fields of characteristic two. Avoiding supersingular curves, an elliptic curve over $\mathbb{F}_{2^n}$ can be expressed by the short Weierstraß form

$$E_{a_2, a_6} : y^2 + xy = x^3 + a_2x^2 + a_6 \quad .$$

From Eqs. (V.4) and (V.2), we see that the addition of two points requires one inversion, four multiplies, and one squaring. Note that the multiplication $(x_1 + x_2)(x_1 + x_2 + a_2)$ can also be evaluated as $(x_1 + x_2)^2 + a_2(x_1 + x_2)$.

Over the large prime field $\mathbb{F}_p$ (with $p > 3$), the Weierstraß equation simplifies to

$$E_{a_4, a_6} : y^2 = x^3 + a_4x + a_6 \quad .$$

The addition of two points then requires one inversion, three multiplies, and two squarings. As inversion in a field of large characteristic is a relatively costly operation, projective representations of points are preferred [102]. A projective point $P = (X_P : Y_P : Z_P)$ on the curve satisfies the (projective) Weierstraß equation

$$E_{a_4, a_6} : Y^2Z = X^3 + a_4XZ^2 + a_6Z^3$$

and, when $Z_P \neq 0$, corresponds to the affine point $P = (X_P/Z_P, Y_P/Z_P)$. The notation $(X : Y : Z)$ represents equivalent classes: $(X : Y : Z)$ and $(X' : Y' : Z')$ are equivalent projective representations if and only if there exists a non-zero field element θ such that $X = \theta X'$, $Y = \theta Y'$, and $Z = \theta Z'$. The only point at infinity (i.e., with $Z = 0$) is the identity element $\mathcal{O} = (0 : 1 : 0)$, and the inverse of $P_1 = (X_1 : Y_1 : Z_1)$ is $-P_1 = (X_1 : -Y_1 : Z_1)$. The sum of two points $P_1 = (X_1 : Y_1 : Z_1)$ and $P_2 = (X_2 : Y_2 : Z_2)$ (with $P_1, P_2 \neq \mathcal{O}$ and $P_1 \neq -P_2$) is given by $P_3 = (X_3 : Y_3 : Z_3)$ with

$$X_3 = 2FW, \quad Y_3 = R(G - 2W) - L^2, \quad Z_3 = 2F^3,$$

where $U_1 = X_1Z_2$, $U_2 = X_2Z_1$, $S_1 = Y_1Z_2$, $S_2 = Y_2Z_1$, $Z = Z_1Z_2$, $T = U_1 + U_2$, $M = S_1 + S_2$, $R = T^2 - U_1U_2 + a_4Z^2$, $F = ZM$, $L = MF$, $G = TL$, and $W = R^2 - G$. Therefore, adding two points with the unified formulæ (Eq. (V.4)) requires 17 multiplies plus 1 multiplication by constant a_4 . When $a_4 = -1$, we may write $R = (T - Z)(T + Z) - U_1U_2$ and the number of multiplies decreases to 16 [51].

V.2.2. Other Parameterizations. Every elliptic curve is isomorphic to a Weierstraß form. Parameterizations other than the Weierstraß form may lead to faster unified point addition formulæ. This was independently suggested by Joye and Quisquater [188] and by Liardet and Smart [220]. We illustrate the topic with elliptic curves defined over a large prime field $\mathbb{F}_p$, $p > 3$.

V.2.2.1. Hessian Form. Most elliptic curves containing a copy of $\mathbb{Z}/3\mathbb{Z}$ (and thus with a group order a multiple of 3) can be expressed by the (projective) Hessian form [309]¹

$$H : X^3 + Y^3 + Z^3 = 3DXYZ \quad . \quad (\text{V.5})$$

Let $P_1 = (X_1 : Y_1 : Z_1)$ and $P_2 = (X_2 : Y_2 : Z_2)$ denote points on the Hessian curve H . The inverse of P_1 is $-P_1 = (Y_1 : X_1 : Z_1)$, and the sum $P_3 = P_1 + P_2$ is defined as $(X_3 : Y_3 : Z_3)$ with

$$X_3 = X_2Y_1^2Z_2 - X_1Y_2^2Z_1, \quad Y_3 = X_1^2Y_2Z_2 - X_2^2Y_1Z_1, \quad Z_3 = X_2Y_2Z_1^2 - X_1Y_1Z_2^2$$

when $P_1 \neq P_2$, and

$$X_3 = Y_1(X_1^3 - Z_1^3), \quad Y_3 = X_1(Z_1^3 - Y_1^3), \quad Z_3 = Z_1(Y_1^3 - X_1^3)$$

otherwise. The formulæ for adding two (distinct) points and for doubling a point appear to be different. However, the doubling operation can be rewritten in terms of a (general) addition operation. This is stated in the following lemma.

LEMMA V.1 ([188]). *If $P_1 = (X_1 : Y_1 : Z_1)$ is a point on an Hessian elliptic curve, then*

$$[2](X_1 : Y_1 : Z_1) = (Z_1 : X_1 : Y_1) + (Y_1 : Z_1 : X_1) \quad .$$

Furthermore, we have $(Z_1 : X_1 : Y_1) \neq (Y_1 : Z_1 : X_1)$.

The correctness of the above lemma is easily verified. Let $T_1 = (0 : -1 : 1)$ and $T_2 = -T_1 = (-1 : 0 : 1)$ denote the two points of order 3 on H . We have $[2](X_1 : Y_1 : Z_1) = ((X_1 : Y_1 : Z_1) + T_1) + ((X_1 : Y_1 : Z_1) + T_2) = (Z_1 : X_1 : Y_1) + (Y_1 : Z_1 : X_1)$. Moreover, $(Z_1 : X_1 : Y_1) = (Y_1 : Z_1 : X_1)$ implies $T_1 = T_2$, a contradiction.

V.2.2.2. Jacobi Form. When an elliptic curve contains a copy of $\mathbb{Z}/2\mathbb{Z}$ (which implies that its group order is a multiple of 2), it can be represented by the (extended) Jacobi form [27]

$$J : Y^2 = \epsilon X^4 - 2\delta X^2 Z^2 + Z^4 \quad . \quad (\text{V.6})$$

¹The construction given in [309] assumes that $p \equiv 2 \pmod{3}$.

Let $P_1 = (X_1 : Y_1 : Z_1)$ and $P_2 = (X_2 : Y_2 : Z_2)$ denote points on the Jacobi curve J . The inverse of P_1 is $-P_1 = (-X_1 : Y_1 : Z_1)$, and the sum $P_3 = P_1 + P_2$ is defined as $(X_3 : Y_3 : Z_3)$ with

$$\begin{aligned} X_3 &= X_1 Z_1 Y_2 + Y_1 X_2 Z_2, \quad Z_3 = (Z_1 Z_2)^2 - \epsilon (X_1 X_2)^2, \\ Y_3 &= (Z_3 + 2\epsilon (X_1 X_2)^2)(Y_1 Y_2 - 2\delta X_1 X_2 Z_1 Z_2) + \\ &\quad 2\epsilon X_1 X_2 Z_1 Z_2 (X_1^2 Z_2^2 + Z_1^2 X_2^2). \end{aligned}$$

It is worth noting the same formula applies for adding (distinct) points or for doubling a point.

Finally, when the curve contains a copy of $(\mathbb{Z}/2\mathbb{Z}) \times (\mathbb{Z}/2\mathbb{Z})$, the Jacobi curve equation (V.6) can be rescaled, in most cases, to the value $\epsilon = 1$ [27]. This corresponds exactly to the Jacobi curves initially suggested by Liardet and Smart [220] but represented as the intersection of two quadrics.²

The different costs for point addition formulæ are summarized in Table V.1. Symbols M and C , respectively, stand for the cost of field multiplication and multiplication by a constant. The costs given for the Weierstraß form consider the short curve equation $Y^2 Z = X^3 + a_4 X Z^2 + a_6 Z^3$.

TABLE V.1. Point Addition for Elliptic Curves over $\mathbb{F}_p$, $p > 3$

Parameterization	Cost	Cofactor
Weierstraß form [51] (with unified formulæ)	$17M + 1C$ (general case) $16M + 1C$ ($a_4 = -1$)	—
Hessian form [188]	$12M$	$h \propto 3$
Extended Jacobi form [27]	$13M + 3C$ (general case) $13M + 1C$ ($\epsilon = 1$)	$h \propto 2$ $h \propto 4$
$\cap$ of 2 quadrics [220]	$16M + 1C$	$h \propto 4$

Elliptic curve standards ([IEEE 1363, FIPS 186.2, SECG]) recommend the use of elliptic curves with group order $\#E = h \cdot q$, where q is a prime and the cofactor h is ≤ 4 .

V.2.3. Dummy Operations. Point addition formulæ need not be strictly equivalent. It suffices that the different point additions cannot be distinguished by side-channel analysis. This can be achieved by inserting dummy (field) operations. Such a technique is of particular interest when the costs for adding two (distinct) points and for doubling a point are similar, as is the case for elliptic curves defined over binary fields [76].

Consider the (non-supersingular) elliptic curve over $\mathbb{F}_{2^n}$ given by the short Weierstraß equation

$$E_{a_2, a_6} : y^2 + xy = x^3 + a_2 x^2 + a_6.$$

²The construction given in [220, 27] works whenever $p \equiv 3 \pmod{4}$ and with a $7/8$ probability when $p \equiv 1 \pmod{4}$.

The slope λ given in Eq. (V.3) becomes

$$\lambda = \frac{y_1 + y_2}{x_1 + x_2} \quad \text{when } x_1 \neq x_2, \text{ and } \lambda = x_1 + \frac{y_1}{x_1} \quad \text{when } x_1 = x_2.$$

In terms of field operations, the expression for λ is almost the same for adding or doubling points. The next algorithm (Algorithm V.1) details how inserting two dummy (field) additions allows us to double a point in a way similar to the addition of two distinct points. We assume that registers contain field elements and that registers T_0, T_1, T_2, T_3 are initialized with the coordinates of points P_1 and P_2 being added.

REMARK. *It is implicitly assumed [as well as in all algorithms presented in this chapter] that the loading/storing of random values from different registers is indistinguishable. A possible solution is presented in [234] when this latter assumption is not verified. See also [176] and Section V.4.*

ALGORITHM V.1: Point Addition (with Dummy Operations)

INPUT: Points $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2) \in E_{a_2, a_6}(\mathbb{F}_{2^n})$, $P_1 \neq -P_2$;
 registers $(T_0, T_1) \leftarrow P_1$ and $(T_2, T_3) \leftarrow P_2$.

OUTPUT: The sum $P_1 + P_2$ or $[2]P_1$.

Addition: $P_1 \leftarrow P_1 + P_2$

Doubling: $P_1 \leftarrow [2]P_1$

- | | | | |
|------------------------------------|-----------------------|--------------------------------|-------------------------|
| 1. $T_0 \leftarrow T_0 + T_2$ | $(= x_1 + x_2)$ | $T_5 \leftarrow T_0 + T_2$ | (fake). |
| 2. $T_1 \leftarrow T_1 + T_3$ | $(= y_1 + y_2)$ | $T_5 \leftarrow T_2 + T_5$ | $(= x_1)$. |
| 3. $T_4 \leftarrow T_1/T_0$ | $(= \lambda)$ | $T_4 \leftarrow T_1/T_0$ | $(= y_1/x_1)$. |
| 4. $T_0 \leftarrow T_0 + T_4$ | | $T_4 \leftarrow T_0 + T_4$ | $(= \lambda)$. |
| 5. $T_5 \leftarrow T_4^2$ | $(= \lambda^2)$ | $T_0 \leftarrow T_4^2$ | $(= \lambda^2)$. |
| 6. $T_5 \leftarrow T_5 + a_2$ | $(= \lambda^2 + a_2)$ | $T_0 \leftarrow T_0 + a_2$ | $(= \lambda^2 + a_2)$. |
| 7. $T_0 \leftarrow T_0 + T_5$ | $(= x_3)$ | $T_0 \leftarrow T_0 + T_4$ | $(= x_3)$. |
| 8. $T_1 \leftarrow T_0 + T_3$ | $(= x_3 + y_2)$ | $T_1 \leftarrow T_0 + T_1$ | $(= x_3 + y_1)$. |
| 9. $T_5 \leftarrow T_0 + T_2$ | $(= x_2 + x_3)$ | $T_5 \leftarrow T_0 + T_5$ | $(= x_1 + x_3)$. |
| 10. $T_4 \leftarrow T_4 \cdot T_5$ | | $T_4 \leftarrow T_4 \cdot T_5$ | . |
| 11. $T_1 \leftarrow T_1 + T_4$ | $(= y_3)$ | $T_1 \leftarrow T_1 + T_4$ | $(= y_3)$. |
| 12. Return (T_0, T_1) . | | | |
-

Hence, Algorithm V.1 only requires one inversion, two multiplies and one squaring for adding or doubling points, in an indistinguishable fashion, on a (non-supersingular) elliptic curve over $\mathbb{F}_{2^n}$.

V.3. Regular Point Multiplication Algorithms

Point addition formulæ may have different side-channel traces, provided that they do not leak information about k in the point multiplication algorithm used for evaluating $Q = [k]P$. For binary algorithms, this implies that the processing of bits “0” and bits “1” of multiplier k are indistinguishable.

V.3.1. Classical Algorithms. The usual trick for removing the conditional branching in the double-and-add algorithm (i.e., the additively written square-and-multiply algorithm) consists in performing a dummy point addition when multiplier bit k_j is zero. As a result, each iteration appears as a point doubling followed by a point addition [87], as illustrated in Algorithm IV.3. Note that a NAF-representation for k in Algorithm IV.3 does not improve the performances. An algorithm using a NAF-representation is presented in [171].

Another popular point multiplication algorithm was developed by Montgomery [255] as a means for speeding up the ECM factoring method on a special form of elliptic curves.

ALGORITHM V.2: Montgomery Point Multiplication Algorithm

INPUT: Point P and ℓ -bit multiplier $k = \sum_{j=0}^{\ell-1} k_j 2^j$, $k_j \in \{0, 1\}$.

OUTPUT: $Q = [k]P$.

1. $R_0 \leftarrow P$, $R_1 \leftarrow [2]P$.
 2. For $j = \ell - 2$ to 0 by -1 do:
 3. $R_{1-k_j} \leftarrow R_0 + R_1$.
 4. $R_{k_j} \leftarrow [2]R_{k_j}$.
 5. Return R_0 .
-

Algorithm V.2 behaves regularly [225, 264] but seems, at first glance, as costly as the double-and-add-always algorithm. However, it does not need to handle the y -coordinates: the sum of two points whose difference is a known point can be computed without the y -coordinates [255]. Note, that the difference $R_1 - R_0$ remains invariant throughout Algorithm V.2 (and hence is equal to P). Further, the y -coordinate of a point R_0 can be recovered from a point P , the x -coordinate of R_0 , and the x -coordinate $R_0 + P$. The y -coordinate of $Q = [k]P$ can thus also be recovered when only x -coordinates are used in the Montgomery point multiplication algorithm. The formulæ for (general) elliptic curves over fields of characteristic $\neq 2, 3$ can be found in [51, 178, 120] and in [225] over binary fields.

Montgomery point multiplication is particularly suited to elliptic curves over binary fields in a normal basis representation [ECC, Section II.2.2] as each iteration globally amounts to only six (field) multiplications. The algorithm is due to López and Dahab [225].

ALGORITHM V.3: Montgomery Multiplication (for Binary Curves)

INPUT: Point $P = (x_P, y_P) \in E_{a_2, a_6}(\mathbb{F}_{2^n})$ and
 ℓ -bit multiplier $k = \sum_{j=0}^{\ell-1} k_j 2^j$, $k_j \in \{0, 1\}$.

OUTPUT: $x_Q = x([k]P)$.

1. $Z_0 \leftarrow 1$, $X_0 \leftarrow x_P$, $Z_1 \leftarrow x_P^2$, $X_1 \leftarrow Z_1^2 + a_6$.
 2. For $j = \ell - 2$ to 0 by -1 do:
 3. $T_0 \leftarrow X_0 \cdot Z_1$, $T_1 \leftarrow X_1 \cdot Z_0$.
 4. $Z_{1-k_j} \leftarrow (T_0 + T_1)^2$, $X_{1-k_j} \leftarrow x_P \cdot Z_{1-k_j} + T_0 \cdot T_1$.
 5. $T_0 \leftarrow Z_{k_j}^2$, $T_1 \leftarrow X_{k_j}^2$.
 6. $Z_{k_j} \leftarrow T_0 \cdot T_1$, $X_{k_j} \leftarrow T_1^2 + a_6 \cdot T_0^2$.
 7. Return X_0/Z_0 .
-

Some applications require both the x - and y -coordinates of point $Q = [k]P$. At the end of Algorithm V.3, registers X_1 and Z_1 , respectively, contain the values of the projective X - and Z -coordinates of point $[k+1]P$. The y -coordinate of Q can then be recovered as

$$y_Q = \frac{(x_Q + x_P)[(x_Q + x_P)(X_1/Z_1 + x_P) + x_P^2 + y_P]}{x_P} + y_P.$$

V.3.2. Atomic Algorithms. The idea behind the double-and-add-always algorithm (Algorithm IV.3) is to remove the conditional branching in the basic double-and-add algorithm so that each iteration appears as a point doubling followed by a point addition. This idea was later generalized and extended by Chevallier-Mames, Ciet and Joye, resulting in the concept of *side-channel atomicity* [76].

There is a better option than the double-and-add-always algorithm when point doubling and point addition are indistinguishable by side-channel analysis (see Section V.2). In this case, the conditional branching can be removed so that the whole point multiplication algorithm appears as a regular repetition of point additions. By doing so, we obtain the atomic double-and-add algorithm [76]. As a side-effect, such an algorithm leaks the Hamming weight of multiplier k . While this is generally not an issue (see however [61]), the Hamming weight can be masked using standard multiplier randomization techniques (see Section V.5).

ALGORITHM V.4: Atomic Double-and-Add Algorithm

INPUT: Point P and ℓ -bit multiplier $k = \sum_{j=0}^{\ell-1} k_j 2^j$, $k_j \in \{0, 1\}$.

OUTPUT: $Q = [k]P$.

1. $R_0 \leftarrow P$, $R_1 \leftarrow P$, $j \leftarrow \ell - 2$, $b \leftarrow 0$.
 2. While $(j \geq 0)$ do:
 3. $R_0 \leftarrow R_0 + R_b$.
 4. $b \leftarrow b \oplus k_j$, $j \leftarrow j + k_j - 1$.
 5. Return R_0 .
-

The above algorithm assumes that (i) XOR-ing k_j with bit b , and (ii) adding k_j to $j - 1$ does not leak information about multiplier bit k_j . If so, again randomization techniques allow us to mask the value of k_j . For example, $b \oplus k_j$ can be evaluated as $b \oplus (k_j \oplus r) \oplus r$ for a random bit r , and $j + k_j - 1$ can be evaluated as $j + (k_j + t) - (t + 1)$ for a random integer t .

If needed, point addition and point subtraction formulæ can be adapted so that they become indistinguishable. It is then advantageous to replace the double-and-add algorithm with the double-and-add/subtract algorithm. When the multiplier k is expressed as a NAF, this results in a 11.11% speed-up factor [256], on average. It is easy to modify Algorithm V.4 to take as input a signed representation for the multiplier k . For a signed digit k_j , $\text{sgn}(k_j)$ indicates whether k_j is negative (i.e., $\text{sgn}(k_j) = 1$ if $k_j < 0$ and 0 otherwise) and $|k_j|$ denotes the absolute value of k_j . To ease the presentation, $\mp_1$ represents a point subtraction and $\mp_0$ represents a point addition (i.e., $P_1 \mp_1 P_2 = P_1 - P_2$ and $P_1 \mp_0 P_2 = P_1 + P_2$).

ALGORITHM V.5: Atomic Double-and-Add/Subtract Algorithm

INPUT: Point P and multiplier $k = 2^\ell + \sum_{j=0}^{\ell-1} k_j 2^j$, $k_j \in \{0, -1, 1\}$.

OUTPUT: $Q = [k]P$.

1. $R_0 \leftarrow P$, $R_1 \leftarrow P$, $j \leftarrow \ell - 1$, $b \leftarrow 0$.
 2. While $(j \geq 0)$ do:
 3. $\sigma \leftarrow b \wedge \text{sgn}(k_j)$.
 4. $R_0 \leftarrow R_0 \mp_\sigma R_b$.
 5. $b \leftarrow b \oplus |k_j|$, $j \leftarrow j + b - 1$.
 6. Return R_0 .
-

Using the indistinguishable point addition formulæ of Algorithm V.1, this yields a very efficient algorithm for (non-supersingular) elliptic curves over binary fields. For elliptic curves over large prime fields, the indistinguishable point addition formulæ are somewhat costly, at least for general elliptic curves (see Table V.1). Rather than considering point addition as an atomic operation, we may look at the field level and express point addition and point

ALGORITHM V.6: Atomic Point Multiplication (over $\mathbb{F}_p$)

INPUT: Point $P = (X_P : Y_P : Z_P)_J \in E_{a_4, a_6}(\mathbb{F}_p)$,
 multiplier $k = 2^\ell + \sum_{j=0}^{\ell-1} k_j 2^j$, $k_j \in \{0, -1, 1\}$, and
 matrices $((u_0)_{r,c})_{0 \leq r \leq 10, 0 \leq c \leq 9}$, $((u_1)_{r,c})_{0 \leq r \leq 10, 0 \leq c \leq 15}$ and $((u_2)_{r,c})_{0 \leq r \leq 10, 0 \leq c \leq 15}$.
OUTPUT: $Q = [k]P$.

1. $T_0 \leftarrow a_4$, $T_1 \leftarrow X_1$, $T_2 \leftarrow Y_1$, $T_3 \leftarrow Z_1$, $T_7 \leftarrow X_1$, $T_8 \leftarrow Y_1$, $T_9 \leftarrow Z_1$.
2. $j \leftarrow \ell - 1$, $b \leftarrow 0$, $s \leftarrow 0$.
3. **While** ($j \geq 0$) **do**:
4. $\sigma \leftarrow b \wedge \text{sgn}(k_j)$.
5. $c \leftarrow s \cdot (c + 1)$, $t \leftarrow b + \sigma$.
6. $T_{(u_t)_{0,c}} \leftarrow T_{(u_t)_{1,c}} \cdot T_{(u_t)_{2,c}}$, $T_{(u_t)_{3,c}} \leftarrow T_{(u_t)_{4,c}} + T_{(u_t)_{5,c}}$.
7. $T_{(u_t)_{6,c}} \leftarrow -T_{(u_t)_{6,c}}$, $T_{(u_t)_{7,c}} \leftarrow T_{(u_t)_{8,c}} + T_{(u_t)_{9,c}}$.
8. $s \leftarrow T_{(u_t)_{10,c}}$.
9. $b \leftarrow (\neg s) \wedge (b \oplus |k_j|)$, $j \leftarrow j + (b \vee s) - 1$.
10. **Return** $(T_1 : T_2 : T_3)_J$.

There are numerous variants of this algorithm, which may be more or less appropriate and efficient for a given architecture.

V.4. Base-Point Randomization Techniques

What makes elliptic curves particularly fruitful for designing countermeasures is the ability to represent elements of the underlying group in many different, randomized ways while keeping a good computational efficiency. This section presents a variety of strategies built on the mathematical structure of the curves, which lead to efficient and simple techniques for randomizing (the representation of) base-point P in the computation of $Q = [k]P$.

V.4.1. Point Blinding. The method is analogous to Chaum's blind signature scheme for RSA [65]. Point P to be multiplied is "blinded" by adding a secret random point R for which the value of $S = [k]R$ is known. The point multiplication, $Q = [k]P$, is done by computing the point $[k](P + R)$ and subtracting S to get Q . Points R and $S = [k]R$ can be initially stored inside the device and refreshed at each new execution by computing $R \leftarrow [r]R$ and $S \leftarrow [r]S$, where r is a (small) random generated at each new execution [87, 205]. As R is secret, the representation of point $P^* = P + R$ is unknown in the computation of $Q^* = [k]P^*$.

V.4.2. Randomized Projective Representations. In projective coordinates, points are not uniquely represented. For example, in homogeneous coordinates, the triplets $(\theta X_P : \theta Y_P : \theta Z_P)$ with any $\theta \neq 0$ represent the same point; and similarly in Jacobian coordinates, the triplets $(\theta^2 X_P : \theta^3 Y_P : \theta Z_P)_J$

for any $\theta \neq 0$ represent the same point. Other projective representations are described in [77, 85].

Before each new execution of the point multiplication algorithm for computing $Q = [k]P$, the projective representation of input point P is randomized with a random non-zero value θ [87]. This makes it no longer possible to predict any specific bit in the binary representation of P .

The randomization can also occur prior to each point addition or at random during the course of the point multiplication algorithm.

V.4.3. Randomized Elliptic Curve Isomorphisms. Point $P = (x_P, y_P)$ on elliptic curve E is randomized as $P^* = \phi(P)$ on $E^* = \phi(E)$, for a random curve isomorphism ϕ . Then $Q = [k]P$ is evaluated as $Q = \phi^{-1}([k]P^*)$ [190], or schematically,

$$\begin{array}{ccc} P \in E(\mathbb{K}) & \xrightarrow{\text{mult. by } k \text{ map}} & Q = [k]P \in E(\mathbb{K}) \\ \phi \downarrow & & \uparrow \phi^{-1} \\ P^* \in E^*(\mathbb{K}) & \xrightarrow{\text{mult. by } k \text{ map}} & Q^* = [k]P^* \in E^*(\mathbb{K}) \end{array} .$$

More specifically, for a random $v \in \mathbb{K} \setminus \{0\}$, point $P = (x_P, y_P)$ on the Weierstraß curve E given by Eq. (V.1) is mapped to point $P^* = (v^2 x_P, v^3 y_P)$ on the isomorphic curve

$$E^* : y^2 + (va_1)xy + (v^3 a_3)y = x^3 + (v^2 a_2)x^2 + (v^4 a_4)x + v^6 a_6 .$$

The next step consists of computing $Q^* = [k]P^*$ on E^* . If $Q^* = \mathcal{O}$, then $Q = \mathcal{O}$; otherwise, if $Q^* = (x_{Q^*}, y_{Q^*})$, then $Q = (v^{-2} x_{Q^*}, v^{-3} y_{Q^*})$.

V.4.4. Randomized Field Isomorphisms. Here, given a point P on an elliptic curve E defined over a field $\mathbb{K}$, a random field isomorphism $\kappa : \mathbb{K} \rightarrow \mathbb{K}^*$ is applied to P and E to get point $P^* = \kappa(P)$ on $E^* = \kappa(E)$. Then, point multiplication $Q = [k]P$ is evaluated as $\kappa^{-1}([k]P^*)$ [190].

V.5. Multiplier Randomization Techniques

Analogously to Section V.4, this section reviews several strategies for randomizing the computation of $Q = [k]P$ but with a randomized (representation of) multiplier k .

V.5.1. Multiplier Blinding. Let $\text{ord}(P)$ denote the order of point $P \in E(\mathbb{K})$. We obviously have

$$[k]P = [k + r \text{ord}(P)]P$$

for any r . As a consequence, the multiplier k can be blinded as $k^* = k + r \text{ord}(P)$ for a random r and $Q = [k]P$ can be computed as $Q = [k^*]P$. Alternatively, since by Lagrange's Theorem the order of an element always divides the order of its group, we can randomize the multiplier k as $k^* =$

$k + r \#E$ for a random r , where $\#E$ denotes the group order of elliptic curve E [87, 205].

V.5.2. Multiplier Splitting. The multiplier k can also be decomposed into two (or several) shares. The idea of splitting the data was already abstracted in [64] as a general countermeasure against side-channel attacks.

Using an additive splitting, k is written as $k = k_1^* + k_2^*$, where $k_1^* = r$ and $k_2^* = k - r$ for a random r . Then $Q = [k]P$ is evaluated as $Q = [k_1^*]P + [k_2^*]P$ [82]. To hide all the bits of multiplier k , the size of r should be comparable to that of k . A possible direction to shorten the size of r consists of decomposing k as $k = k_1^* r + k_2^*$ with $k_1^* = \lfloor k/r \rfloor$ and $k_2^* = k \bmod r$, and evaluating Q as $Q = [k_1^*]R + [k_2^*]P$ with $R = [r]P$.

We can also use a multiplicative splitting and evaluate $Q = [k]P$ as $Q = [kr^{-1}](rP)$ for a random r invertible modulo $\text{ord}(P)$ [329].

V.5.3. Frobenius Endomorphism. The Koblitz elliptic curves (a.k.a. anomalous binary curves or ABC's) [202] are the curves K_0 and K_1 over $\mathbb{F}_{2^n}$ given by

$$K_{a_2} : y^2 + xy = x^3 + a_2x^2 + 1$$

with $a_2 \in \{0, 1\}$. On such curves, the Frobenius endomorphism φ maps a point $P = (x, y)$ to $\varphi(P) = (x^2, y^2) \in K_{a_2}$. Therefore, in the computation of $Q = [k]P$, the scalar k can be written as a Frobenius expansion since multiplication by k is an endomorphism and $\mathbb{Z} \subseteq \mathbb{Z}[\varphi] \subseteq \text{End}(K_{a_2})$.

The ring $\mathbb{Z}[\varphi]$ is a Euclidean domain with respect to the norm $N(r + s\varphi) = r^2 + (-1)^{1-a_2}rs + 2s^2$. Furthermore, as $N(\varphi) = 2$, every element $r + s\varphi$ in $\mathbb{Z}[\varphi]$ can be written as a φ -NAF, that is,

$$r + s\varphi = \sum_i k_i \varphi^i \quad \text{with } k_i \in \{-1, 0, 1\} \text{ and } k_i \cdot k_{i+1} = 0.$$

Unfortunately, the so-obtained Frobenius expansion is roughly twice the length of the usual balanced binary expansion, and so, even if the evaluation of φ is very fast, it is not clear that the resulting method is faster. This drawback was looped in [237, 316] with the following observation. We obviously have $\varphi^n = 1$ and thus $Q = [k']P$ with $k' = k \bmod (\varphi^n - 1)$. As $N(\varphi^n - 1) = \#K_{a_2}(\mathbb{F}_{2^n}) \approx 2^n$ by Hasse's Theorem, the φ -NAF expression of k' , $k' = \sum_i k'_i \varphi^i$, would have a length approximatively equal to that of the (usual) NAF expression of k .

This yields the following method for randomizing the multiplier k on Koblitz curves [190]. For any nonzero $\rho \in \mathbb{Z}[\varphi]$, $k \bmod (\rho(\varphi^n - 1))$ acts identically on a point $P \in K_{a_2}(\mathbb{F}_{2^n})$ and so $Q = [k]P$ can be evaluated as

$$\sum_i \kappa_i^* \varphi^i(P),$$

where $\sum_i \kappa_i^* \varphi^i$ is the φ -NAF expression of κ^* and $\kappa^* = k \bmod (\rho(\varphi^n - 1))$ for a (short) non-zero random $\rho \in \mathbb{Z}[\varphi]$.

For cryptographic applications, n is chosen prime and so $\#K_0(\mathbb{F}_{2^n}) = 4q$ or $\#K_1(\mathbb{F}_{2^n}) = 2q$ for a prime q . In that case, when working in the subgroup of order q , k can be reduced modulo $\rho(\varphi^n - 1)/(\varphi - 1)$ [211]. See also [161] for further randomization techniques dedicated to Koblitz curve cryptosystems and [79] for similar randomization techniques on elliptic curves having non-trivial efficiently computable endomorphisms.

V.6. Preventing Side-Channel Analysis

The general (software) methodology for preventing side-channel leakage is a two-step process. The first step consists in making SPA-like attacks impossible and the second step in making DPA-like attacks impossible.

SPA-like attacks are inapplicable when it is not possible to relate secret data with a single execution of the cryptoalgorithm. For elliptic curve cryptosystems, this implies that the value of k (or a part thereof) cannot be recovered by monitoring the processing of $Q = [k]P$. Various techniques towards this goal were presented in Sections V.2 and V.3.

To successively mount a DPA-like attack, an adversary should be able to predict any relevant bit. Therefore, a proper randomization of the inputs to the cryptoalgorithm yields an implementation secure against DPA-like attacks. Several randomization techniques for base-point P and multiplier k in the point multiplication $Q = [k]P$ were presented in Sections V.4 and V.5, respectively. The randomization should of course be effective to prevent refined attacks [151, 6] (cf. Section IV.5.4). As a general rule of thumb, both P and k should be randomized in the computation of $Q = [k]P$.

Part 3

Mathematical Foundations

CHAPTER VI

Advances in Point Counting

F. Vercauteren

The preferred method for generating elliptic curves suitable for cryptographic applications is to choose a large finite field and to randomly select curves over that field until one is found with nearly prime group order.

Let $\overline{E}$ be an elliptic curve over $\mathbb{F}_q$, with $q = p^n$. The number of $\mathbb{F}_q$ -rational points $\#\overline{E}(\mathbb{F}_q)$ satisfies the relation $\#\overline{E}(\mathbb{F}_q) = q + 1 - t$, where $t = \text{Tr}(\overline{F})$ is the trace of the Frobenius endomorphism $\overline{F} : \overline{E}(\overline{\mathbb{F}}_q) \rightarrow \overline{E}(\overline{\mathbb{F}}_q) : (x, y) \mapsto (x^q, y^q)$. By Hasse's theorem [307, Theorem V.1.1] we have $|t| \leq 2\sqrt{q}$, so it suffices to compute t modulo B with $B > 4\sqrt{q}$.

In 1985, Schoof [292] described the first polynomial time algorithm to compute $\#\overline{E}(\mathbb{F}_q)$ using an l -adic approach. The key idea of Schoof's algorithm is to compute t modulo sufficiently many primes l such that $\prod l \geq B$. The time complexity of Schoof's algorithm is $O(\log^{3\mu+2} q)$ and the space complexity amounts to $O(\log^3 q)$, with μ a constant such that multiplication of two m -bit integers can be computed in $O(m^\mu)$ time. For example: $\mu = 2$ for classical multiplication, $\mu = \log_2 3$ for Karatsuba multiplication [194] and $\mu = 1 + \varepsilon$ with $\varepsilon \in \mathbb{R}_{>0}$ for Schönhage–Strassen [291] multiplication.

Several improvements mainly due to Elkies [115] and Atkin [11] resulted in a heuristically estimated time complexity of $O(\log^{2\mu+2} q)$ and space complexity of $O(\log^2 q)$. Further work by Couveignes [90, 91] and Lercier [217] extended this so called SEA algorithm to work in small characteristic. A nice introduction to these l -adic algorithms can be found in Chapter VII of the first volume [ECC].

At the end of 1999, Satoh [285] proposed a completely different strategy based on p -adic methods. The main idea is to lift the curve and the Frobenius endomorphism to a p -adic ring and to recover t modulo p^m with $p^m > 4\sqrt{q}$ directly from the lifted data. For fixed p , the time complexity of Satoh's algorithm is $O(n^{2\mu+1})$ and the space complexity is $O(n^3)$. Since the O -constant of the time complexity grows as $O(p^2 \log^\mu p)$, Satoh's algorithm is only practical for small p .

The purpose of this chapter is to explain the mathematical background of Satoh's algorithm and to present an overview of the more recent improvements culminating in an algorithm by Harley that runs in $O(n^{2\mu} \log n)$ time using $O(n^2)$ space for fixed p .

VI.1. p -adic Fields and Extensions

All p -adic point counting algorithms lift the Frobenius endomorphism to a ring $\mathbb{Z}_q$ whose reduction modulo p is isomorphic to $\mathbb{F}_q$. In this section we give a brief overview of the arithmetic in $\mathbb{Z}_q$. Further details can be found in the books by Koblitz [201] and Serre [295].

Recall that the field of p -adic numbers $\mathbb{Q}_p$ consists of power series

$$\alpha = \sum_{i=\nu}^{\infty} a_i p^i,$$

with $\nu \in \mathbb{Z}$, $0 \leq a_i < p$ and $a_\nu \neq 0$. By definition, the valuation of α is $\text{ord}_p(\alpha) = \nu$ and the norm is $|\alpha|_p = p^{-\nu}$. The ring of p -adic integers or the valuation ring of $\mathbb{Q}_p$ is defined as

$$\mathbb{Z}_p = \{\alpha \in \mathbb{Q}_p \mid \text{ord}_p(\alpha) \geq 0\}.$$

Note that for every $m \in \mathbb{N}$ we have $\mathbb{Z}_p/(p^m \mathbb{Z}_p) \cong \mathbb{Z}/(p^m \mathbb{Z})$, so $\mathbb{Z}_p$ modulo p simply is $\mathbb{F}_p$. Every $\beta \in \mathbb{Z}_p$ with $\alpha \equiv \beta \pmod{p^m}$ is called an approximation of α to precision m .

Let $f(t)$ be a monic polynomial of degree n with coefficients in $\mathbb{Z}_p$ and assume that $\bar{f}(t) \equiv f(t) \pmod{p}$ is an irreducible polynomial over $\mathbb{F}_p$. Note that this implies that $f(t)$ itself is irreducible over $\mathbb{Q}_p$. The algebraic extension $\mathbb{Q}_q = \mathbb{Q}_p(\theta)$ with θ a root of $f(t)$ is called an unramified extension of degree n of $\mathbb{Q}_p$. Arithmetic in $\mathbb{Q}_q$ thus reduces to polynomial arithmetic in $\mathbb{Q}_p[t]/(f(t))$. Let $\alpha = \sum_{i=0}^{n-1} \alpha_i t^i$ with $\alpha_i \in \mathbb{Q}_p$. Then the valuation of α is defined as $\text{ord}_p(\alpha) = \min_i \text{ord}_p(\alpha_i)$ and the norm is $|\alpha|_p = p^{-\text{ord}_p(\alpha)}$. The valuation ring $\mathbb{Z}_q$ is defined as $\mathbb{Z}_p[t]/(f(t))$ or equivalently $\mathbb{Z}_q = \{\alpha \in \mathbb{Q}_q \mid \text{ord}_p(\alpha) \geq 0\}$.

The Galois group $\text{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)$ is isomorphic to $\text{Gal}(\mathbb{F}_q/\mathbb{F}_p)$. Indeed, every automorphism $\Lambda \in \text{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)$ induces an automorphism of $\mathbb{F}_q$ by reduction modulo p . Since $\bar{f}(t)$ is square-free, this mapping is an isomorphism. The Galois group $\text{Gal}(\mathbb{F}_q/\mathbb{F}_p)$ is cyclic and generated by the p th power Frobenius automorphism

$$\bar{\sigma} : \mathbb{F}_q \rightarrow \mathbb{F}_q : x \mapsto x^p.$$

The unique automorphism $\Sigma \in \text{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)$ with $\Sigma \equiv \bar{\sigma} \pmod{p}$ is called the Frobenius substitution on $\mathbb{Q}_q$. Note that, unlike $\bar{\sigma}$, the Frobenius substitution is not simply p th powering. Since Σ is an $\mathbb{Z}_p$ -automorphism, i.e., is the identity on $\mathbb{Z}_p$, we have $\Sigma(\alpha) = \sum_{i=0}^{n-1} \alpha_i \Sigma(t)^i$. The element $\Sigma(t) \in \mathbb{Z}_q$ can be computed using the Newton iteration $x_{k+1} \leftarrow x_k - f(x_k)/f'(x_k)$ starting with $x_0 = t^p$.

Given a representation $\mathbb{F}_q \cong \mathbb{F}_p[t]/(\bar{f}(t))$, there are infinitely many polynomials $f(t) \in \mathbb{Z}_p[t]$ with $f(t) \equiv \bar{f}(t) \pmod{p}$. However, from an algorithmic point of view, there are two lifts of $\bar{f}(t)$ that lead to efficient arithmetic. Note that $\bar{f}(t)$ is normally chosen to be sparse, e.g. a trinomial or pentanomial in characteristic 2, such that reduction modulo $\bar{f}(t)$ is very efficient.

- Let $\bar{f}(t) = \sum_{i=0}^n \bar{f}_i t^i$ with $\bar{f}_i \in \mathbb{F}_p$ for $0 \leq i \leq n$. To preserve the sparsity of $\bar{f}(t)$, a first natural choice is to define $f(t) = \sum_{i=0}^n f_i t^i$ with f_i the unique integer between 0 and $p-1$ such that $f_i \equiv \bar{f}_i \pmod{p}$. The reduction modulo $f(t)$ of a polynomial of degree $\leq 2(n-1)$ then only takes $n(w-1)$ multiplications of a $\mathbb{Z}_p$ -element by a small integer and nw subtractions in $\mathbb{Z}_p$ where w is the number of non-zero coefficients of $\bar{f}(t)$.
- Since $\mathbb{F}_q$ is the splitting field of the polynomial $t^q - t$, the polynomial $\bar{f}(t)$ divides $t^q - t$. To preserve the simple Galois structure of $\mathbb{F}_q$, a second natural choice is to define $f(t)$ as the unique polynomial over $\mathbb{Z}_p$ with $f(t)|t^q - t$ and $f(t) \equiv \bar{f}(t) \pmod{p}$. Every root $\theta \in \mathbb{Z}_q$ of $f(t)$ clearly is a $(q-1)$ -th root of unity, therefore $\Sigma(\theta)$ is also a $(q-1)$ -th root of unity. Since $\Sigma(\theta) \equiv \theta^p \pmod{p}$, we conclude that $\Sigma(\theta) = \theta^p$ or by abuse of notation $\Sigma(t) = t^p$.

VI.2. Satoh's Algorithm

The basic idea of Satoh's algorithm is to lift the curve $\bar{E}/\mathbb{F}_q$ to a curve $\mathcal{E}/\mathbb{Q}_q$ such that the Frobenius endomorphism $\bar{F} \in \text{End}(\bar{E})$ also lifts to an endomorphism $\mathcal{F} \in \text{End}(\mathcal{E})$. Since $\text{Tr}(\mathcal{F}) = \text{Tr}(\bar{F})$, we can recover the number of points as $\#\bar{E}(\mathbb{F}_q) = q + 1 - \text{Tr}(\mathcal{F})$. Furthermore, $\mathcal{E}$ and $\mathcal{F}$ are defined over a field of characteristic zero, which allows us to compute $\text{Tr}(\mathcal{F})$ modulo p^m for any precision m by analysing the action of $\mathcal{F}$ on an invariant differential of $\mathcal{E}$.

VI.2.1. The Canonical Lift of an Ordinary Elliptic Curve. Among the many possible lifts of $\bar{E}$ from $\mathbb{F}_q$ to $\mathbb{Q}_q$, there is essentially only one that admits a lift of the Frobenius endomorphism $\bar{F}$.

DEFINITION VI.1. *The canonical lift $\mathcal{E}$ of an ordinary elliptic curve $\bar{E}$ over $\mathbb{F}_q$ is an elliptic curve over $\mathbb{Q}_q$ that satisfies:*

- *the reduction modulo p of $\mathcal{E}$ equals $\bar{E}$,*
- *$\text{End}(\bar{E}) \cong \text{End}(\mathcal{E})$ as a ring.*

Deuring [101] has shown that the canonical lift $\mathcal{E}$ always exists and is unique up to isomorphism. Furthermore, a theorem by Lubin, Serre and Tate [227] provides an effective algorithm to compute the j -invariant of $\mathcal{E}$ given the j -invariant of $\bar{E}$.

THEOREM VI.2 (Lubin-Serre-Tate). *Let $\bar{E}$ be an ordinary elliptic curve over $\mathbb{F}_q$ with j -invariant $j(\bar{E}) \in \mathbb{F}_q \setminus \mathbb{F}_{p^2}$. Denote with Σ the Frobenius substitution on $\mathbb{Q}_q$ and with $\Phi_p(X, Y)$ the p th modular polynomial [ECC, pp. 50–55]. Then the system of equations*

$$\Phi_p(X, \Sigma(X)) = 0 \quad \text{and} \quad X \equiv j(\bar{E}) \pmod{p}, \quad (\text{VI.1})$$

has a unique solution $J \in \mathbb{Z}_q$, which is the j -invariant of the canonical lift $\mathcal{E}$ of $\bar{E}$ (defined up to isomorphism).

The hypothesis $j(\overline{E}) \notin \mathbb{F}_{p^2}$ in Theorem VI.2 is necessary to ensure that a certain partial derivative of Φ_p does not vanish modulo p and guarantees the uniqueness of the solution of equation (VI.1). The case $j(\overline{E}) \in \mathbb{F}_{p^2}$ can be handled very easily using the Weil conjectures [346]: since $j(\overline{E}) \in \mathbb{F}_{p^2}$ there exists an elliptic curve $\overline{E}'$ defined over $\mathbb{F}_{p^e}$, with $e = 1$ or $e = 2$, that is isomorphic to $\overline{E}$ over $\mathbb{F}_q$. Let $t_k = p^{ek} + 1 - \#\overline{E}'(\mathbb{F}_{p^{ek}})$. Then $t_{k+1} = t_1 t_k - p^e t_{k-1}$ with $t_0 = 2$ and therefore $\#\overline{E}(\mathbb{F}_q) = p^n + 1 - t_{n/e}$. So in the remainder of this chapter we will assume that $j(\overline{E}) \notin \mathbb{F}_{p^2}$ and in particular that $\overline{E}$ is ordinary [307, Theorem V.3.1].

VI.2.2. Isogeny Cycles. Lifting the Frobenius endomorphism $\overline{F}$ directly is not efficient since the degree of $\overline{F}$ is q . However, $q = p^n$, so the Frobenius endomorphism can be written as the composition of n isogenies of degree p . Let $\overline{\sigma} : \overline{E} \rightarrow \overline{E}^\sigma : (x, y) \mapsto (x^p, y^p)$ be the p th power Frobenius isogeny, where $\overline{E}^\sigma$ is the curve obtained by raising each coefficient of $\overline{E}$ to the p th power. Repeatedly applying $\overline{\sigma}$ gives rise to the following cycle:

$$\overline{E}_0 \xrightarrow{\overline{\sigma}_0} \overline{E}_1 \xrightarrow{\overline{\sigma}_1} \dots \xrightarrow{\overline{\sigma}_{n-2}} \overline{E}_{n-1} \xrightarrow{\overline{\sigma}_{n-1}} \overline{E}_0,$$

with $\overline{E}_i = \overline{E}^{\overline{\sigma}^i}$ and $\overline{\sigma}_i : \overline{E}_i \rightarrow \overline{E}_{i+1} : (x, y) \mapsto (x^p, y^p)$. Composing these isogenies, we can express the Frobenius endomorphism as

$$\overline{F} = \overline{\sigma}_{n-1} \circ \overline{\sigma}_{n-2} \circ \dots \circ \overline{\sigma}_0.$$

Instead of lifting $\overline{E}$ separately, Satoh lifts the whole cycle $(\overline{E}_0, \overline{E}_1, \dots, \overline{E}_{n-1})$ simultaneously leading to the diagram

$$\begin{array}{ccccccc} \mathcal{E}_0 & \xrightarrow{\Sigma_0} & \mathcal{E}_1 & \xrightarrow{\Sigma_1} & \dots & \xrightarrow{\Sigma_{n-2}} & \mathcal{E}_{n-1} & \xrightarrow{\Sigma_{n-1}} & \mathcal{E}_0 \\ \pi \downarrow & & \pi \downarrow & & & & \pi \downarrow & & \pi \downarrow \\ \overline{E}_0 & \xrightarrow{\overline{\sigma}_0} & \overline{E}_1 & \xrightarrow{\overline{\sigma}_1} & \dots & \xrightarrow{\overline{\sigma}_{n-2}} & \overline{E}_{n-1} & \xrightarrow{\overline{\sigma}_{n-1}} & \overline{E}_0, \end{array} \quad (\text{VI.2})$$

with $\mathcal{E}_i$ the canonical lift of $\overline{E}_i$ and Σ_i the corresponding lift of $\overline{\sigma}_i$. The existence of a lift of $\overline{\sigma}_i$ is not trivial and follows from the following theorem which can be found in Messing [246, Corollary 3.4].

THEOREM VI.3. *Let $\overline{E}_1, \overline{E}_2$ be ordinary elliptic curves over $\mathbb{F}_q$ and $\mathcal{E}_1, \mathcal{E}_2$ their respective canonical liftings. Then $\text{Hom}(\overline{E}_1, \overline{E}_2) \cong \text{Hom}(\mathcal{E}_1, \mathcal{E}_2)$.*

VI.2.3. Computing the Canonical Lift. The theorem of Lubin, Serre and Tate implies that the j -invariants of the $\mathcal{E}_i$ satisfy

$$\Phi_p(j(\mathcal{E}_i), j(\mathcal{E}_{i+1})) = 0 \quad \text{and} \quad j(\mathcal{E}_i) \equiv j(\overline{E}_i) \pmod{p},$$

for $i = 0, \dots, n-1$. Define $\Theta : \mathbb{Z}_q^n \rightarrow \mathbb{Z}_q^n$ by

$$\Theta(x_0, x_1, \dots, x_{n-1}) = (\Phi_p(x_0, x_1), \Phi_p(x_1, x_2), \dots, \Phi_p(x_{n-1}, x_0)).$$

Then clearly we have $\Theta(j(\mathcal{E}_{n-1}), j(\mathcal{E}_{n-2}), \dots, j(\mathcal{E}_0)) = (0, 0, \dots, 0)$. Using a multivariate Newton iteration on Θ , we can lift the cycle of j -invariants $(j(\overline{E}_{n-1}), j(\overline{E}_{n-2}), \dots, j(\overline{E}_0))$ to $\mathbb{Z}_q^n$ with arbitrary precision. The iteration is given by

$$(x_0, x_1, \dots, x_{n-1}) \leftarrow (x_0, x_1, \dots, x_{n-1}) - ((D\Theta)^{-1}\Theta)(x_0, x_1, \dots, x_{n-1}),$$

with $(D\Theta)(x_0, \dots, x_{n-1})$ the Jacobian matrix

$$\begin{pmatrix} \Phi'_p(x_0, x_1) & \Phi'_p(x_1, x_0) & 0 & \cdots & 0 \\ 0 & \Phi'_p(x_1, x_2) & \Phi'_p(x_2, x_1) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \Phi'_p(x_{n-1}, x_{n-2}) \\ \Phi'_p(x_0, x_{n-1}) & 0 & 0 & \cdots & \Phi'_p(x_{n-1}, x_0) \end{pmatrix},$$

where $\Phi'_p(X, Y)$ denotes the partial derivative with respect to X . Note that $\Phi'_p(Y, X)$ is the partial derivative of $\Phi_p(X, Y)$ with respect to Y since $\Phi_p(X, Y)$ is symmetric.

The p th modular polynomial satisfies the Kronecker relation

$$\Phi_p(X, Y) \equiv (X^p - Y)(X - Y^p) \pmod{p},$$

and since $j(\overline{E}_i) \notin \mathbb{F}_{p^2}$ and $j(\overline{E}_{i+1}) = j(\overline{E}_i)^p$, we have

$$\begin{cases} \Phi'_p(j(\overline{E}_{i+1}), j(\overline{E}_i)) & \equiv j(\overline{E}_i)^{p^2} - j(\overline{E}_i) \not\equiv 0 \pmod{p}, \\ \Phi'_p(j(\overline{E}_i), j(\overline{E}_{i+1})) & \equiv j(\overline{E}_i)^p - j(\overline{E}_i)^p \equiv 0 \pmod{p}. \end{cases}$$

The above equations imply that $(D\Theta)(x_0, \dots, x_{n-1}) \pmod{p}$ is a diagonal matrix with non-zero diagonal elements. Therefore, the Jacobian matrix is invertible over $\mathbb{Z}_q$ and thus $\delta = ((D\Theta)^{-1}\Theta)(x_0, x_1, \dots, x_{n-1}) \in \mathbb{Z}_q^n$. Note that we can simply apply Gauss elimination to solve

$$(D\Theta)(x_0, \dots, x_{n-1})\delta = \Theta(x_0, \dots, x_{n-1})$$

since the diagonal elements are all invertible. Using row operations we move the bottom left element $\Phi'_p(x_0, x_{n-1})$ towards the right. After k row operations this element becomes

$$(-1)^k \Phi'_p(x_0, x_{n-1}) \prod_{i=0}^{k-1} \frac{\Phi'_p(x_{i+1}, x_i)}{\Phi'_p(x_i, x_{i+1})},$$

which clearly is divisible by p^k since $\Phi'_p(x_{i+1}, x_i) \equiv 0 \pmod{p}$. This procedure is summarized in Algorithm VI.1.

ALGORITHM VI.1: Lift_j-Invariants

INPUT: Cycle $j_i \in \mathbb{F}_q \setminus \mathbb{F}_{p^2}$ with $\Phi_p(j_i, j_{i+1}) \equiv 0 \pmod{p}$ for $0 \leq i < n$ and precision m .
OUTPUT: Cycle $J_i \in \mathbb{Z}_q$ with $\Phi_p(J_i, J_{i+1}) \equiv 0 \pmod{p^m}$ and $J_i \equiv j_i \pmod{p}$ for all $0 \leq i < n$.

1. If $m = 1$ then
2. For $i = 0$ to $n - 1$ do $J_i \leftarrow j_i$.
3. Else
4. $m' \leftarrow \lceil \frac{m}{2} \rceil$, $M \leftarrow m - m'$.
5. $(J_0, \dots, J_{n-1}) \leftarrow \text{Lift_j_Invariants}((j_0, \dots, j_{n-1}), m')$.
6. For $i = 0$ to $n - 2$ do:
7. $t \leftarrow \Phi'_p(J_i, J_{i+1})^{-1} \pmod{p^M}$.
8. $D_i \leftarrow t \Phi'_p(J_{i+1}, J_i) \pmod{p^M}$.
9. $P_i \leftarrow t((\Phi_p(J_i, J_{i+1}) \pmod{p^m})/p^{m'}) \pmod{p^M}$.
10. $R \leftarrow \Phi'_p(J_0, J_{n-1}) \pmod{p^M}$.
11. $S \leftarrow (((\Phi_p(J_{n-1}, J_0) \pmod{p^m}))/p^{m'}) \pmod{p^M}$.
12. For $i = 0$ to $\min(M, n - 2)$ do:
13. $S \leftarrow S - R P_i \pmod{p^M}$.
14. $R \leftarrow -R D_i \pmod{p^M}$.
15. $R \leftarrow R + \Phi'_p(J_{n-1}, J_0) \pmod{p^M}$.
16. $P_{n-1} \leftarrow S R^{-1} \pmod{p^M}$.
17. For $i = n - 2$ to 0 by -1 do $P_i \leftarrow P_i - D_i P_{i+1} \pmod{p^M}$.
18. For $i = 0$ to $n - 1$ do $J_i \leftarrow J_i - p^{m'} P_i \pmod{p^m}$.
19. Return $(J_0, \dots, J_{n-1})$.

As shown in Chapter III of the first volume [ECC], we can assume that either $\overline{E}$ or its quadratic twist is given by an equation of the form

$$\begin{aligned}
 p = 2 & : y^2 + xy = x^3 + a, & j(\overline{E}) &= 1/a, \\
 p = 3 & : y^2 = x^3 + x^2 + a, & j(\overline{E}) &= -1/a, \\
 p > 5 & : y^2 = x^3 + 3ax + 2a, & j(\overline{E}) &= 1728a/(1+a).
 \end{aligned}$$

Once the j -invariant $j(\mathcal{E})$ of the canonical lift of $\overline{E}$ is computed, a Weierstraß model for $\mathcal{E}$ is given by

$$\begin{aligned}
 p = 2 & : y^2 + xy = x^3 + 36\alpha x + \alpha, & \alpha &= 1/(1728 - j(\mathcal{E})), \\
 p = 3 & : y^2 = x^3 + x^2/4 + 36\alpha x + \alpha, & \alpha &= 1/(1728 - j(\mathcal{E})), \\
 p > 5 & : y^2 = x^3 + 3\alpha x + 2\alpha, & \alpha &= j(\mathcal{E})/(1728 - j(\mathcal{E})).
 \end{aligned}$$

Note that the above models have the correct j -invariant $j(\mathcal{E})$ and reduce to $\overline{E}$ modulo p .

VI.2.4. The Trace of Frobenius. The canonical lift $\mathcal{E}$ of an ordinary elliptic curve $\overline{E}$ over $\mathbb{F}_q$ has the property that $\text{End}(\overline{E}) \cong \text{End}(\mathcal{E})$. Let $\mathcal{F}$ be the image of the Frobenius endomorphism $\overline{F}$ under this ring isomorphism; then clearly $\text{Tr}(\overline{F}) = \text{Tr}(\mathcal{F})$. Furthermore, since $\mathbb{Q}_q$ has characteristic zero, the exact value of $\text{Tr}(\overline{F})$ can be computed, and not just modulo p . The following proposition gives an easy relation between the trace of an endomorphism $f \in \text{End}(E)$ and its action on an invariant differential of E .

PROPOSITION VI.4 (Sato). *Let E be an elliptic curve over $\mathbb{Q}_q$ having good reduction modulo p and let $f \in \text{End}(E)$ of degree d . Let ω be an invariant differential on E and let $f^*(\omega) = c\omega$ be the action of f on ω . Then*

$$\text{Tr}(f) = c + \frac{d}{c}.$$

Note that if we apply the above proposition to the lifted Frobenius endomorphism $\mathcal{F}$, we get $\text{Tr}(\mathcal{F}) = b + q/b$ with $\mathcal{F}^*(\omega) = b\omega$. Since $\overline{E}$ is ordinary it follows that either b or q/b is a unit in $\mathbb{Z}_q$. However, $\overline{F}$ is inseparable and thus $b \equiv 0 \pmod{p}$, which implies that $b \equiv 0 \pmod{q}$, since q/b has to be a unit in $\mathbb{Z}_q$. So if we want to compute $\text{Tr}(\mathcal{F}) \pmod{p^m}$, we would need to determine $b \pmod{p^{n+m}}$. Furthermore, it turns out to be quite difficult to compute b directly. As will become clear later, we would need to know $\text{Ker}(\Sigma_i)$, which is a subgroup of $\mathcal{E}_i[p]$ of order p . However, since $\text{Ker}(\overline{\sigma}_i)$ is trivial we cannot use a simple lift of $\text{Ker}(\overline{\sigma}_i)$ to $\text{Ker}(\Sigma_i)$, but would need to factor the p -division polynomial of $\mathcal{E}_i$.

To avoid these problems, Satoh works with the Verschiebung $\overline{V}$, i.e., the dual isogeny of $\overline{F}$, which is separable since $\overline{E}$ is ordinary and thus $\text{Ker}(\overline{V})$ can be easily lifted to $\text{Ker}(\mathcal{V})$, with $\mathcal{V}$ the image of $\overline{V}$ under the ring isomorphism $\text{End}(\overline{E}) \cong \text{End}(\mathcal{E})$. Furthermore, the trace of an endomorphism equals the trace of its dual, so we have $\text{Tr}(\mathcal{F}) = \text{Tr}(\mathcal{V}) = c + q/c$ with $\mathcal{V}^*(\omega) = c\omega$ and c a unit in $\mathbb{Z}_q$. Diagram (VI.2) shows that $\mathcal{V} = \widehat{\Sigma}_0 \circ \widehat{\Sigma}_1 \circ \cdots \circ \widehat{\Sigma}_{n-1}$ and therefore we can compute c from the action of $\widehat{\Sigma}_i$ on an invariant differential ω_i of $\mathcal{E}_i$ for $i = 0, \dots, n-1$. More precisely, take $\omega_i = \Sigma^i(\omega)$ for $0 \leq i < n$ and let c_i be defined by

$$\widehat{\Sigma}_i^*(\omega_i) = c_i \omega_{i+1}.$$

Then $c = \prod_{0 \leq i < n} c_i$. Since $\overline{V}$ is separable, c will be non-zero modulo p and we conclude

$$\text{Tr}(\overline{F}) \equiv \prod_{0 \leq i < n} c_i \pmod{q}.$$

Since all commutative squares in diagram (VI.2) are conjugates of each other, we can also recover the trace of Frobenius as the norm of c_0 , i.e.,

$$\text{Tr}(\overline{F}) \equiv N_{\mathbb{Q}_q/\mathbb{Q}_p}(c_0) \pmod{q}.$$

VI.2.5. Computing the c_i . The final step in Satoh's algorithm is to compute the coefficients c_i using Vélú's formulae [332], which need the equations for $\mathcal{E}_i$ and $\mathcal{E}_{i+1}$ and the kernel of $\widehat{\Sigma}_i$. Consider the following diagram:

$$\begin{array}{ccc} \mathcal{E}_{i+1} & \xrightarrow{\widehat{\Sigma}_i} & \mathcal{E}_i \\ & \searrow \nu_i \quad \nearrow \lambda_i & \\ & \mathcal{E}_{i+1}/\text{Ker}(\widehat{\Sigma}_i) & \end{array} \quad (\text{VI.3})$$

Given $\text{Ker}(\widehat{\Sigma}_i)$, Satoh uses Vélú's formulae [332] to compute an equation for the curve $\mathcal{E}_i/\text{Ker}(\widehat{\Sigma}_i)$ and the isogeny ν_i . Since ν_i and $\widehat{\Sigma}_i$ are both separable, $\text{Ker}(\nu_i) = \text{Ker}(\widehat{\Sigma}_i)$ and $\deg(\nu_i) = \deg(\widehat{\Sigma}_i)$, there exists an isomorphism $\lambda_i : \mathcal{E}_{i+1}/\text{Ker}(\widehat{\Sigma}_i) \rightarrow \mathcal{E}_i$ that makes the above diagram commutative. Due to Vélú's construction, the action of ν_i on the invariant differential is trivial, i.e., $\nu_i^*(\omega_{i+1,K}) = \omega_{i+1}$ with $\omega_{i+1,K}$ the invariant differential on $\mathcal{E}_{i+1}/\text{Ker}(\widehat{\Sigma}_i)$. Therefore it is sufficient to compute the action of λ_i on ω_i .

Note that $\text{Ker}\widehat{\Sigma}_i$ is a subgroup of order p of $\mathcal{E}_{i+1}[p]$. Let $H_i(x)$ be

$$H_i(x) = \prod_{P \in (\text{Ker}\widehat{\Sigma}_i \setminus \{\mathcal{O}\})/\pm} (x - x(P));$$

then $H_i(x)$ divides the p -division polynomial $\Psi_{p,i+1}(x)$ of $\mathcal{E}_{i+1}$. To find the correct factor of $\Psi_{p,i+1}(x)$ Satoh proves the following lemma.

LEMMA VI.5 (Satoh). *Let $p \geq 3$. Then $\text{Ker}\widehat{\Sigma}_i = \mathcal{E}_{i+1}[p] \cap \mathcal{E}_{i+1}(\mathbb{Z}_q^{\text{ur}})$, with $\mathbb{Z}_q^{\text{ur}}$ the valuation ring of the maximal unramified extension $\mathbb{Q}_q^{\text{ur}}$ of $\mathbb{Q}_q$.*

The above lemma implies that $H_i(x) \in \mathbb{Z}_q[x]$ is the unique monic polynomial of degree $(p-1)/2$ that divides $\Psi_{p,i+1}(x)$ and such that $H_i(x) \pmod{p}$ is square-free. Since $\overline{E}_i$ is ordinary, $\text{Ker}\widehat{\sigma}_i = \overline{E}_{i+1}[p]$ and $\Psi_{p,i+1}(x) \pmod{p}$ has inseparable degree p . Therefore, $\delta_i H_i(x)^p \equiv \Psi_{p,i+1}(x) \pmod{p}$. This implies that we cannot apply Hensel's lemma since the polynomials $H_i(x) \pmod{p}$ and $\Psi_{p,i+1}(x)/H_i(x) \pmod{p}$ are not coprime. To this end, Satoh devised a modified Hensel lifting [285, Lemma 2.1], which has quadratic convergence.

LEMMA VI.6 (Satoh). *Let $p \geq 3$ be a prime and $\Psi(x) \in \mathbb{Z}_q[x]$ satisfying $\Psi'(x) \equiv 0 \pmod{p}$ and $\Psi'(x) \not\equiv 0 \pmod{p^2}$. Let $h(x) \in \mathbb{Z}_q[x]$ be a monic polynomial such that*

1. $h(x) \pmod{p}$ is square-free and coprime to $(\Psi'(x)/p) \pmod{p}$,
2. $\Psi(x) \equiv q(x)h(x) \pmod{p^{m+1}}$.

Then the polynomial

$$H(x) = h(x) + \left(\frac{\Psi(x)}{\Psi'(x)} h'(x) \pmod{h(x)} \right)$$

satisfies $H(x) \equiv h(x) \pmod{p^m}$ and $\Psi(x) \equiv Q(x)H(x) \pmod{p^{2m+1}}$.

ALGORITHM VI.2: Lift_Kernel

INPUT: The p -division polynomial $\Psi_p(x)$ of an elliptic curve $\mathcal{E}$ over $\mathbb{Z}_q/(p^m\mathbb{Z}_q)$, precision m .

OUTPUT: $H(x) = \prod_{P \in (\text{Ker}\widehat{\Sigma}_i \setminus \{\mathcal{O}\})/\pm} (x - x(P)) \pmod{p^{m-1}}$.

1. If $m = 1$ then
 2. $H(x) \leftarrow h(x)$ monic with $\Psi_p(x) \equiv \delta h(x)^p \pmod{p}$.
 3. Else
 4. $m' \leftarrow \lceil \frac{m-1}{2} \rceil$.
 5. $H(x) \leftarrow \text{Lift_Kernel}(\Psi_p(x), m')$.
 6. $H(x) \leftarrow H(x) + \left(\frac{H'(x)\Psi_p(x)}{\Psi_p'(x)} \pmod{H(x)} \right) \pmod{p^m}$.
 7. Return $H(x)$.
-

For $p > 3$, $\mathcal{E}_{i+1}$ can be defined by the equation $y^2 = x^3 + A_{i+1}x + B_{i+1}$. Using Vélú's formulae, Satoh [285, Proposition 4.3] shows that $\mathcal{E}_{i+1}/\text{Ker}(\widehat{\Sigma}_i)$ is given by the equation $y^2 = x^3 + \alpha_{i+1}x + \beta_{i+1}$ with

$$\begin{aligned}\alpha_{i+1} &= (6 - 5p)A_{i+1} - 30(h_{i,1}^2 - 2h_{i,2}), \\ \beta_{i+1} &= (15 - 14p)B_{i+1} - 70(-h_{i,1}^3 + 3h_{i,1}h_{i,2} - 3h_{i,3}) + 42A_{i+1}h_{i,1},\end{aligned}$$

where $h_{i,k}$ denotes the coefficient of $x^{(p-1)/2-k}$ in $H_i(x)$ and we define $h_{i,k} = 0$ for $(p-1)/2 - k < 0$.

Given the above Weierstraß model for $\mathcal{E}_{i+1}/\text{Ker}(\widehat{\Sigma}_i)$ we can now compute the isomorphism λ_i to $\mathcal{E}_i : y^2 = x^3 + A_i x + B_i$. The only change of variables preserving the form of these equations is $\lambda_i : (x, y) \rightarrow (u_i^2 x, u_i^3 y)$ with

$$u_i^2 = \frac{\alpha_{i+1}}{\beta_{i+1}} \frac{B_i}{A_i}.$$

The action of λ_i on ω_i is given by $\lambda_i^*(\omega_i) = u_i^{-1} \omega_{i+1,K}$, and therefore

$$c_i^2 = \frac{\beta_{i+1}}{\alpha_{i+1}} \frac{A_i}{B_i}. \tag{VI.4}$$

Computing $c^2 = \prod_{i=0}^{n-1} c_i^2 = N_{\mathbb{Q}_q/\mathbb{Q}_p}(c_0^2)$ and taking the square root gives the trace of Frobenius up to the sign. As shown in the proof of [307, Theorem V.4.1], we have

$$t \equiv \gamma \gamma^{\bar{\sigma}} \dots \gamma^{\bar{\sigma}^{n-1}} \pmod{p},$$

where γ is the coefficient of x^{p-1} in the polynomial $(x^3 + 3ax + 2a)^{(p-1)/2}$. This finally leads to Algorithm VI.3.

ALGORITHM VI.3: Satoh

INPUT: Elliptic curve $\overline{E}: y^2 = x^3 + ax + b$ over $\mathbb{F}_{p^n}$, $j(\overline{E}) \notin \mathbb{F}_{p^2}$.

OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{p^n})$.

1. $m \leftarrow \lceil \log_p 4 + n/2 \rceil$.
 2. $S \leftarrow 1, T \leftarrow 1$.
 3. $j_0 \leftarrow j_n \leftarrow j(\overline{E})$.
 4. **For** $i = 0$ **to** $n - 2$ **do**:
 5. $j_{i+1} \leftarrow j_i^p$.
 6. $(J_{n-1}, \dots, J_0) \leftarrow \text{Lift_j_Invariants}((j_{n-1}, \dots, j_0), m)$.
 7. **For** $i = 0$ **to** $n - 1$ **do**:
 8. $\gamma \leftarrow J_i / (1728 - J_i) \pmod{p^m}$.
 9. $A \leftarrow 3\gamma \pmod{p^m}, B \leftarrow 2\gamma \pmod{p^m}$.
 10. $\Psi_p(x) \leftarrow p\text{-division polynomial of } y^2 = x^3 + Ax + B$.
 11. $H(x) \leftarrow \text{Lift_Kernel}(\Psi_p(x), m + 1)$.
 12. **For** $j = 1$ **to** 3 **do**:
 13. $h_j \leftarrow \text{Coeff}(H(x), (p - 1)/2 - j)$.
 14. $\alpha \leftarrow (6 - 5p)A - 30(h_1^2 - 2h_2)$.
 15. $\beta \leftarrow (15 - 14p)B - 70(-h_1^3 + 3h_1h_2 - 3h_3) + 42Ah_1$.
 16. $S \leftarrow \beta AS, T \leftarrow \alpha BT$.
 17. $t \leftarrow \text{Sqrt}(S/T, m)$.
 18. $\gamma \leftarrow \text{Coeff}((x^3 + ax + b)^{(p-1)/2}, p - 1)$.
 19. **If** $t \not\equiv \gamma\gamma^\sigma \dots \gamma^{\sigma^{n-1}} \pmod{p}$ **then** $t \leftarrow -t \pmod{p^m}$.
 20. **If** $t^2 > 4p^n$ **then** $t \leftarrow t - p^m$.
 21. **Return** $p^n + 1 - t$.
-

The case $p = 3$ is very similar to the case $p \geq 5$. There are only two minor adaptations: firstly, note that $\text{Ker } \widehat{\Sigma}_i = \{Q, -Q, \mathcal{O}\}$ with Q a 3-torsion point on $\mathcal{E}_{i+1}$ with integral coordinates, so Algorithm VI.2 reduces to a simple Newton iteration on the 3-division polynomial of $\mathcal{E}_{i+1}$; secondly, the Weierstraß equation for $\mathcal{E}_i$ is different from the one for $p \geq 5$, which slightly changes Vélú's formulae. Let x_Q denote the x -coordinate of $Q \in \text{Ker } \widehat{\Sigma}_i$ and let $\mathcal{E}_{i+1}$ be defined by $y^2 = x^3 + x^2/4 + A_{i+1}x + B_{i+1}$. Then $\mathcal{E}_{i+1}/\text{Ker}(\widehat{\Sigma}_i)$ is given by the equation $y^2 = x^3 + x^2/4 + \alpha_{i+1}x + \beta_{i+1}$, with

$$\begin{aligned}\alpha_{i+1} &= -15x_Q^2 - (5/2)x_Q - 4A_{i+1}, \\ \beta_{i+1} &= -49x_Q^3 - (27/2)x_Q^2 - (35A_{i+1} + 1/2)x_Q - A_{i+1} - 27B_{i+1}.\end{aligned}$$

Analogous to the case $p \geq 5$, we conclude that c_i^2 is given by (VI.4) and taking the square root of $c^2 = \prod_{i=0}^{n-1} c_i^2$ determines the trace of Frobenius t up to the sign. Furthermore, since the curve $\overline{E}$ is defined by an equation of the form $y^2 = x^3 + x^2 + a$, the correct sign follows from $t \equiv 1 \pmod{3}$.

For $p = 2$, Lemma VI.5 no longer holds. Indeed, the Newton polygon of the 2-division polynomial shows that there are two non-trivial points in $\mathcal{E}_{i+1}[p] \cap \mathcal{E}_{i+1}(\mathbb{Z}_q^{\text{ur}})$, whereas $\text{Ker } \widehat{\Sigma}_i$ has only one non-trivial point. The main problem in extending Satoh's algorithm to characteristic 2 therefore lies in choosing the correct 2-torsion point. There are two algorithms which are both based on diagram (VI.3). Let $\text{Ker } \widehat{\Sigma} = \langle Q \rangle$; then, since λ is an isomorphism, we conclude $j(\mathcal{E}_{i+1}/\langle Q \rangle) = j(\mathcal{E}_i)$.

The first algorithm to compute Q is due to Skjernaa [308] who gives an explicit formula for the x -coordinate x_Q as a function of $j(\mathcal{E}_i)$ and $j(\mathcal{E}_{i+1})$. Since Q is a 2-torsion point, it follows that $2y_Q + x_Q = 0$. Substituting y_Q in the equation of the curve and using the equality $j(\mathcal{E}_{i+1}/\langle Q \rangle) = j(\mathcal{E}_i)$, Skjernaa deduces an explicit expression for x_Q . A proof of the following proposition can be found in [308, Lemma 4.1].

PROPOSITION VI.7. *Let $Q = (x_Q, y_Q)$ be the non-trivial point in $\text{Ker } \widehat{\Sigma}_{i+1}$ and let $z_Q = x_Q/2$. Then*

$$z_Q = -\frac{j(\mathcal{E}_i)^2 + 195120j(\mathcal{E}_i) + 4095j(\mathcal{E}_{i+1}) + 660960000}{8(j(\mathcal{E}_i)^2 + j(\mathcal{E}_i)(563760 - 512j(\mathcal{E}_{i+1})) + 372735j(\mathcal{E}_{i+1}) + 8981280000)}.$$

Skjernaa shows that the 2-adic valuation of both the numerator and the denominator is 12, so we have to compute $j(\mathcal{E}_i) \pmod{2^{m+12}}$ to recover $z_Q \pmod{2^m}$.

The second algorithm is due to Fouquet, Gaudry and Harley [123] and is based on the fact that $\text{Ker } \widehat{\Sigma}_i = \langle Q \rangle \subset \mathcal{E}_{i+1}[2]$. Let $\mathcal{E}_{i+1}$ be given by the equation $y^2 + xy = x^3 + 36A_{i+1}x + A_{i+1}$ with $A_{i+1} = 1/(1728 - j(\mathcal{E}_{i+1}))$. Since Q is a 2-torsion point, we have $2y_Q + x_Q = 0$ and the x -coordinate x_Q is a zero of the 2-division polynomial $4x^3 + x^2 + 144A_{i+1}x + 4A_{i+1}$. Clearly we have $x_Q \equiv 0 \pmod{2}$, so Fouquet, Gaudry and Harley compute $z_Q = x_Q/2$ as a zero of the modified 2-division polynomial $8z^3 + z^2 + 72A_{i+1}z + A_{i+1}$. The main problem is choosing the correct starting value when considering this equation modulo 8. Using $j(\mathcal{E}_{i+1}/\langle Q \rangle) = j(\mathcal{E}_i)$ they proved that $z \equiv 1/j(\mathcal{E}_i) \pmod{8}$ is the correct starting value giving x_Q .

Vélu's formulae show that $\mathcal{E}_{i+1}/\text{Ker } \widehat{\Sigma}_i$ is given by the Weierstraß equation $y^2 + xy = x^3 + \alpha_{i+1}x + \beta_{i+1}$ with

$$\begin{aligned}\alpha_{i+1} &= -\frac{36}{j(\mathcal{E}_{i+1}) - 1728} - 5\gamma_{i+1}, \\ \beta_{i+1} &= -\frac{1}{j(\mathcal{E}_{i+1}) - 1728} - (1 + 7x_Q)\gamma_{i+1},\end{aligned}$$

where $\gamma_{i+1} = 3x_Q^2 - 36/(j(\mathcal{E}_{i+1}) - 1728) + x_Q/2$. The isomorphism λ_i now has the general form

$$(x, y) \rightarrow (u_i^2x + r_i, u_i^3y + u_i^2s_ix + t_i), \quad (u_i, r_i, s_i, t_i) \in \mathbb{Q}_q^* \times \mathbb{Q}_q^3,$$

but an easy calculation shows that $c_i^2 = u_i^{-2}$. Solving the equations satisfied by (u_i, r_i, s_i, t_i) given in [307, Table 1.2] finally leads to

$$c_i^2 = -\frac{864\beta_i - 72\alpha_i + 1}{48\alpha_i - 1}. \quad (\text{VI.5})$$

The complexity of Algorithm VI.3 directly follows from Hasse's theorem, which states that $|t| \leq 2\sqrt{q}$. Therefore it suffices to lift all data with precision $m \simeq n/2$. Since elements of $\mathbb{Z}_q/(p^m\mathbb{Z}_q)$ are represented as polynomials of degree less than n with coefficients in $\mathbb{Z}/(p^m\mathbb{Z})$, every element takes $O(n^2)$ memory for fixed p . Therefore, multiplication and division in $\mathbb{Z}_q/(p^m\mathbb{Z}_q)$ take $O(n^{2\mu})$ time.

For each curve $\overline{E}_i$ with $0 \leq i < n$ we need $O(1)$ elements of $\mathbb{Z}_q/(p^m\mathbb{Z}_q)$, so the total memory needed is $O(n^3)$ bits. Lifting the cycle of j -invariants to precision m requires $O(\log m)$ iterations. In every iteration the precision of the computations almost doubles, so the complexity is determined by the last iteration, which takes $O(n^{2\mu+1})$ bit-operations. Computing one coefficient c_i^2 requires $O(1)$ multiplications, so to compute all c_i we also need $O(n^{2\mu+1})$ bit-operations.

In conclusion, there exists a deterministic algorithm to compute the number of points on an elliptic curve $\overline{E}$ over a finite field $\mathbb{F}_q$ with $q = p^n$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$, which requires $O(n^{2\mu+1})$ bit-operations and $O(n^3)$ space for fixed p .

VI.2.6. Vercauteren's Algorithm. The first improvement of Satoh's algorithm and its extensions to characteristics 2 and 3 is an algorithm by Vercauteren [334] that only requires $O(n^2)$ space but still runs in $O(n^{2\mu+1})$ time. The basic idea is very simple: the trace of Frobenius t can be computed as $t \equiv \prod_{0 \leq i < n} c_i \pmod{q}$ and the c_i only depend on the curves $\mathcal{E}_i$ and $\mathcal{E}_{i+1}$. So the main problem of Satoh's algorithm is that it lifts all j -invariants simultaneously, instead of lifting one j -invariant at a time.

Let $J_i \equiv j(\mathcal{E}_i) \pmod{p^m}$. Then $J_{i+1} \equiv j(\mathcal{E}_{i+1}) \pmod{p^m}$ can be computed using a univariate Newton iteration on $\Phi_p(X, J_i)$. This iteration is given by

$$Z \leftarrow Z - \frac{\Phi_p(Z, J_i)}{\frac{\partial \Phi_p}{\partial X}(Z, J_i)}, \quad (\text{VI.6})$$

starting from $j(\overline{E}_{i+1}) \equiv j(\mathcal{E}_{i+1}) \pmod{p}$ as an initial approximation. Note that since $\Phi_p(X, Y)$ satisfies the Kronecker relation, $\frac{\partial \Phi_p}{\partial X}(J_{i+1}, J_i)$ will be a unit in $\mathbb{Z}_q$. The following proposition shows that the iteration (VI.6) has a quite remarkable property: let J_{i+1} be the zero of $\Phi_p(X, J_i)$; then J_{i+1} equals $j(\mathcal{E}_{i+1})$ up to precision $m+1$ and not just up to precision m as one would expect.

PROPOSITION VI.8. *Let $\mathbb{Q}_q$ be an unramified extension of $\mathbb{Q}_p$ and denote with $\mathbb{Z}_q$ its valuation ring. Let $g \in \mathbb{Z}_q[X, Y]$ and assume that $x_0, y_0 \in \mathbb{Z}_q$ satisfy*

$$g(x_0, y_0) \equiv 0 \pmod{p}, \quad \frac{\partial g}{\partial X}(x_0, y_0) \not\equiv 0 \pmod{p}, \quad \frac{\partial g}{\partial Y}(x_0, y_0) \equiv 0 \pmod{p}.$$

Then the following properties hold:

1. *For every $y \in \mathbb{Z}_q$ with $y \equiv y_0 \pmod{p}$ there exists a unique $x \in \mathbb{Z}_q$ such that $x \equiv x_0 \pmod{p}$ and $g(x, y) = 0$.*
2. *Let $y' \in \mathbb{Z}_q$ with $y \equiv y' \pmod{p^m}$, $m \geq 1$ and let $x' \in \mathbb{Z}_q$ be the unique element with $x' \equiv x_0 \pmod{p}$ and $g(x', y') = 0$. Then x' satisfies $x' \equiv x \pmod{p^{m+1}}$.*

Repeatedly applying this proposition immediately leads to the following algorithm:

ALGORITHM VI.4: Lift_First_j_Invariant

INPUT: A j -invariant $j \in \mathbb{F}_{p^n} \setminus \mathbb{F}_{p^2}$ and a precision m .

OUTPUT: $J \in \mathbb{Z}_q$ with $J \equiv j^{p^{m-1}} \pmod{p}$ and $\Phi_p(J, \Sigma(J)) \equiv 0 \pmod{p^m}$.

1. $J \leftarrow j \pmod{p}$.
 2. **For** $i = 2$ **to** m **do**:
 3. $J \leftarrow \text{Newton_Iteration}(\Phi_p(X, J), J^p \pmod{p}, i)$.
 4. **Return** J .
-

To obtain an $O(n^2)$ space algorithm, we simply interleave the lifting of the j -invariants with the computation of c in Step 5 of Algorithm VI.3.

Note that the convergence of Algorithm VI.4 is only linear, i.e., for every iteration of the **For**-loop, the precision increases by one. However, once we have computed one j -invariant with sufficient precision, we can switch to the Newton iteration (VI.6), which converges quadratically.

In conclusion, there exists a deterministic algorithm to compute the number of points on an elliptic curve $\overline{E}$ over a finite field $\mathbb{F}_q$ with $q = p^n$ and $j(\overline{E}) \notin \mathbb{F}_{p^2}$, which requires $O(n^{2\mu+1})$ bit-operations and $O(n^2)$ space for fixed p .

VI.3. Arithmetic Geometric Mean

In a letter to Gaudry and Harley, Mestre [247] described a very elegant algorithm based on the Arithmetic Geometric Mean (AGM) to count the number of points on ordinary curves of genus 1 and 2 over $\mathbb{F}_{2^n}$. Carls [60] and Kohel [208] showed how to extend this algorithm to higher characteristics. In this section we give a detailed description of both the bivariate and univariate versions of the AGM. Note that the AGM algorithm for elliptic curves is protected by a U.S. patent [160] filed by Harley and Mestre.

For $a_0, b_0 \in \mathbb{R}$ with $a_0 \geq b_0 > 0$, define the AGM iteration for $k \in \mathbb{N}$ as

$$(a_{k+1}, b_{k+1}) = \left(\frac{a_k + b_k}{2}, \sqrt{a_k b_k} \right).$$

Then $b_k \leq b_{k+1} \leq a_{k+1} \leq a_k$ and $0 \leq a_{k+1} - b_{k+1} \leq (a_k - b_k)/2$; therefore $\lim_{k \rightarrow \infty} a_k = \lim_{k \rightarrow \infty} b_k$ exists. This common value is called the AGM of a_0 and b_0 and is denoted by $\text{AGM}(a_0, b_0)$. An easy calculation shows that

$$\frac{a_k}{b_k} - 1 \leq \frac{a_0 - b_0}{2^k b_k} \leq \frac{1}{2^k} \left(\frac{a_0}{b_0} - 1 \right),$$

so after a logarithmic number of steps we have $a_k/b_k = 1 + \varepsilon_k$ with $\varepsilon_k < 1$. The Taylor series expansion of $1/\sqrt{1 + \varepsilon_k}$ shows that convergence even becomes quadratic:

$$\frac{a_{k+1}}{b_{k+1}} = \frac{a_k + b_k}{2\sqrt{a_k b_k}} = \frac{2 + \varepsilon_k}{2\sqrt{1 + \varepsilon_k}} = 1 + \frac{\varepsilon_k^2}{8} - \frac{\varepsilon_k^3}{8} + \frac{15\varepsilon_k^4}{128} - \frac{7\varepsilon_k^5}{64} + O(\varepsilon_k^6). \quad (\text{VI.7})$$

VI.3.1. The AGM Isogeny and the Canonical Lift. Let $\mathbb{Q}_q$ be a degree n unramified extension of $\mathbb{Q}_2$ with valuation ring $\mathbb{Z}_q$ and residue field $\mathbb{F}_q$. Denote by $\sqrt{c}$ for $c \in 1 + 8\mathbb{Z}_q$ the unique element $d \in 1 + 4\mathbb{Z}_q$ with $d^2 = c$. For elements $a, b \in \mathbb{Z}_q$ with $a/b \in 1 + 8\mathbb{Z}_q$, $a' = (a + b)/2$ and $b' = b\sqrt{a/b}$ also belong to $\mathbb{Z}_q$ and $a'/b' \in 1 + 8\mathbb{Z}_q$. Furthermore, if $a, b \in 1 + 4\mathbb{Z}_q$, then also $a', b' \in 1 + 4\mathbb{Z}_q$. However, as can be seen from equation (VI.7), the AGM sequence will converge if and only if $a/b \in 1 + 16\mathbb{Z}_q$. For $a/b \in 1 + 8\mathbb{Z}_q$, the AGM sequence will not converge at all.

Let $a, b \in 1 + 4\mathbb{Z}_q$ with $a/b \in 1 + 8\mathbb{Z}_q$ and $E_{a,b}$ the elliptic curve defined by $y^2 = x(x - a^2)(x - b^2)$. Note that $E_{a,b}$ is not a minimal Weierstraß model and that its reduction modulo 2 is singular. The following lemma gives an isomorphism from $E_{a,b}$ to a minimal model.

LEMMA VI.9. *Let $a, b \in 1 + 4\mathbb{Z}_q$ with $a/b \in 1 + 8\mathbb{Z}_q$ and $E_{a,b}$ the elliptic curve defined by $y^2 = x(x - a^2)(x - b^2)$. Then the isomorphism*

$$(x, y) \mapsto \left(\frac{x - ab}{4}, \frac{y - x + ab}{8} \right)$$

transforms $E_{a,b}$ in $y^2 + xy = x^3 + rx^2 + sx + t$ with

$$\begin{cases} r = \frac{-a^2 + 3ab - b^2 - 1}{4}, \\ s = \frac{-a^3b + 2a^2b^2 - ab^3}{8}, \\ t = \frac{-a^4b^2 + 2a^3b^3 - a^2b^4}{64}. \end{cases} \quad (\text{VI.8})$$

Furthermore, $r \in 2\mathbb{Z}_q$, $s \in 8\mathbb{Z}_q$, $t \equiv -\left(\frac{a-b}{8}\right)^2 \pmod{16}$ and the equation defines a minimal Weierstraß model.

The next proposition shows that the AGM iteration constructs a sequence of elliptic curves all of which are 2-isogenous. This will provide the missing link between the AGM iteration and the canonical lift.

PROPOSITION VI.10. Let $a, b \in 1 + 4\mathbb{Z}_q$ with $a/b \in 1 + 8\mathbb{Z}_q$ and $E_{a,b}$ the elliptic curve defined by the equation $y^2 = x(x-a^2)(x-b^2)$. Let $a' = (a+b)/2$, $b' = \sqrt{ab}$ and $E_{a',b'} : y^2 = x(x-a'^2)(x-b'^2)$. Then $E_{a,b}$ and $E_{a',b'}$ are 2-isogenous. The isogeny is given by

$$\phi : E_{a,b} \rightarrow E_{a',b'} : (x, y) \mapsto \left(\frac{(x+ab)^2}{4x}, y \frac{(x-ab)(x+ab)}{8x^2} \right),$$

and the kernel of ϕ is $\langle (0, 0) \rangle$. Furthermore, the action of ϕ on the invariant differential dx/y is

$$\phi^* \left(\frac{dx}{y} \right) = 2 \frac{dx}{y}.$$

Let $\overline{E}$ be an ordinary elliptic curve defined by $y^2 + xy = x^3 + c$ with $c \in \mathbb{F}_q^*$ and let $\mathcal{E}$ be its canonical lift. Take any $r \in \mathbb{Z}_q$ such that $r^2 \equiv c \pmod{2}$ and let $a_0 = 1 + 4r$ and $b_0 = 1 - 4r$. Then Lemma VI.9 shows that E_{a_0, b_0} is isomorphic to a lift of $\overline{E}$ to $\mathbb{Z}_q$ and thus $j(E_{a_0, b_0}) \equiv j(\overline{E}) \pmod{2}$.

Let $(a_k, b_k)_{k=0}^\infty$ be the AGM sequence and consider the elliptic curves E_{a_k, b_k} . Proposition VI.10 implies that $\Phi_2(j(E_{a_{k+1}, b_{k+1}}), j(E_{a_k, b_k})) = 0$ and an easy computation shows that $j(E_{a_{k+1}, b_{k+1}}) \equiv j(E_{a_k, b_k})^2 \pmod{2}$. Note that $\Phi_2(X, Y)$ and $(j(E_{a_{k+1}, b_{k+1}}), j(E_{a_k, b_k}))$ satisfy the conditions of Proposition VI.8, so we conclude

$$j(E_{a_k, b_k}) \equiv \Sigma^k(j(\mathcal{E})) \pmod{2^{k+1}}.$$

Although the AGM sequence $(a_k, b_k)_{k=0}^\infty$ itself does not converge, the sequence of elliptic curves $(E_{a_{nk}, b_{nk}})_{k=0}^\infty$ does converge since $\lim_{k \rightarrow \infty} j(E_{a_{nk}, b_{nk}})$ exists and is equal to the j -invariant of the canonical lift of $\overline{E}$.

VI.3.2. Computing the Trace of Frobenius. The AGM not only provides an efficient algorithm to compute the j -invariant of the canonical lift $\mathcal{E}$, but it also leads to an elegant formula for the trace of Frobenius.

Assume we have computed $a, b \in 1 + 4\mathbb{Z}_q$ with $a/b \in 1 + 8\mathbb{Z}_q$ such that $j(E_{a,b}) = j(\mathcal{E})$. Let $(a', b') = ((a+b)/2, \sqrt{ab})$, $\phi : E_{a,b} \rightarrow E_{a',b'}$ the AGM isogeny and $\Sigma : E_{a,b} \rightarrow E_{\Sigma(a), \Sigma(b)}$ the lift of the 2nd-power Frobenius. Then we have the following diagram:

$$\begin{array}{ccc} E_{a,b} & \xrightarrow{\phi} & E_{a',b'} \\ & \searrow \Sigma \quad \swarrow \lambda & \\ & E_{\Sigma(a), \Sigma(b)} & \end{array} \quad (\text{VI.9})$$

The kernel of the Frobenius isogeny $\Sigma : E_{a,b} \rightarrow E_{\Sigma(a), \Sigma(b)}$ is a subgroup of order 2 of

$$E_{a,b}[2] = \{\mathcal{O}, (0, 0), (a^2, 0), (b^2, 0)\}.$$

The isomorphism given in Lemma VI.9 allows us to analyse the reduction of the 2-torsion on a minimal model. An easy calculation shows that $(0, 0)$ is mapped onto $\mathcal{O}$ whereas $(a^2, 0)$ and $(b^2, 0)$ are mapped to $(0, (a-b)/8 \pmod{2})$. Therefore we conclude that $\text{Ker } \Sigma = \{\mathcal{O}, (0, 0)\}$. Proposition VI.10 shows that $\text{Ker } \Sigma = \text{Ker } \phi$, and, since both isogenies are separable, there exists an isomorphism $\lambda : E_{a',b'} \rightarrow E_{\Sigma(a),\Sigma(b)}$ such that $\Sigma = \lambda \circ \phi$. The following proposition shows that this isomorphism has a very simple form.

PROPOSITION VI.11. *Given two elliptic curves $E_{a,b} : y^2 = x(x - a^2)(x - b^2)$ and $E_{c,d} : y'^2 = x'(x' - c^2)(x' - d^2)$ over $\mathbb{Q}_q$ with $a, b, c, d \in 1 + 4\mathbb{Z}_q$ and $a/b, c/d \in 1 + 8\mathbb{Z}_q$, then $E_{a,b}$ and $E_{c,d}$ are isomorphic if and only if $x' = u^2x$ and $y' = u^3y$ with $u^2 = \frac{c^2+d^2}{a^2+b^2}$. Furthermore, $(\frac{a}{b})^2 = (\frac{c}{d})^2$ or $(\frac{a}{b})^2 = (\frac{d}{c})^2$.*

Let $\omega = dx/y$ and $\omega' = dx'/y'$ be invariant differentials on $E_{a,b}$ and $E_{\Sigma(a),\Sigma(b)}$, respectively. Then

$$\Sigma^*(\omega') = (\lambda \circ \phi)^*(\omega') = 2u^{-1}\omega,$$

with $u^2 = \frac{\Sigma(a)^2 + \Sigma(b)^2}{a'^2 + b'^2}$. Define $\zeta = a/b = 1 + 8c$ and $\zeta' = a'/b' = 1 + 8c'$. Then Proposition VI.11 also implies that

$$\zeta'^2 = \Sigma(\zeta)^2 \quad \text{or} \quad \zeta'^2 = \frac{1}{\Sigma(\zeta)^2}.$$

Substituting $\zeta = 1 + 8c$ and $\zeta' = 1 + 8c'$ in the above equation and dividing by 16, we conclude that $c' \equiv \Sigma(c) \pmod{4}$ or $c' \equiv -\Sigma(c) \pmod{4}$. The Taylor expansion of $1 + 8c' = (1 + 4c)/\sqrt{1 + 8c}$ modulo 32 shows that $c' \equiv c^2 \pmod{4}$. Since after the first iteration c itself is a square α^2 modulo 4, and since $\Sigma(\alpha^2) \equiv \alpha^4 \pmod{4}$, we conclude that $\zeta'^2 = \Sigma(\zeta)^2$. Note that this implies

$$\zeta' = \frac{a'}{b'} = \Sigma(\zeta) = \Sigma\left(\frac{a}{b}\right), \quad (\text{VI.10})$$

since $\zeta' \equiv \zeta \equiv 1 \pmod{8}$. Substituting $b'^2 = a'^2 \Sigma(b)/\Sigma(a)^2$ in the expression for u^2 and taking square roots leads to

$$u = \pm \frac{\Sigma(a)}{a'}.$$

Let $(a_k, b_k)_{k=0}^\infty$ be the AGM sequence with $a_0 = a$ and $b_0 = b$ and consider the following diagram where $\mathcal{E}_k = E_{\Sigma^k(a), \Sigma^k(b)}$ and $\Sigma_k : \mathcal{E}_k \rightarrow \mathcal{E}_{k+1}$ the lift of the 2nd-power Frobenius isogeny.

$$\begin{array}{ccccccc} & \xrightarrow{\phi_0} & \xrightarrow{\phi_1} & \xrightarrow{\phi_2} & \cdots & \xrightarrow{\phi_{n-1}} & \\ E_{a,b} & \longrightarrow & E_{a_1,b_1} & \longrightarrow & E_{a_2,b_2} & \longrightarrow & \cdots \longrightarrow E_{a_n,b_n} \\ \text{Id} \downarrow & & \lambda_1 \downarrow & & \lambda_2 \downarrow & & \vdots \downarrow \lambda_n \\ \mathcal{E}_0 & \xrightarrow{\Sigma_0} & \mathcal{E}_1 & \xrightarrow{\Sigma_1} & \mathcal{E}_2 & \xrightarrow{\Sigma_2} & \cdots \xrightarrow{\Sigma_{n-1}} \mathcal{E}_n = \mathcal{E}_0 \end{array}$$

Since $\text{Ker } \Sigma_k \circ \lambda_k = \text{Ker } \phi_k$ for $k \in \mathbb{N}$, we can repeat the same argument as for diagram (VI.9) and find an isomorphism λ_{k+1} such that the square

commutes, i.e., $\Sigma_k = \lambda_{k+1} \circ \phi_k \circ \lambda_k^{-1}$. Since $E_{a,b}$ is isomorphic to the canonical lift $\mathcal{E}$ of $\overline{E}$, we conclude that

$$\mathrm{Tr}(\Sigma_{n-1} \circ \cdots \circ \Sigma_0) = \mathrm{Tr} \mathcal{F} = \mathrm{Tr} \overline{F}.$$

The above diagram shows that $\Sigma_{n-1} \circ \cdots \circ \Sigma_0 = \lambda_n \circ \phi_{n-1} \circ \cdots \circ \phi_0$ and, since ϕ_k acts on the invariant differential ω as multiplication by 2 and λ_n as multiplication by $\pm a_n/a_0$, we conclude that

$$\mathcal{F}^*(\omega) = \pm 2^n \frac{a_n}{a_0}(\omega).$$

The Weil conjectures imply that the product of the roots of the characteristic polynomial of Frobenius is 2^n and thus

$$\mathrm{Tr} \mathcal{F} = \mathrm{Tr} \overline{F} = \pm \frac{a_0}{a_n} \pm 2^n \frac{a_n}{a_0}. \quad (\text{VI.11})$$

Equation (VI.10) implies that $a_{k+1}/a_{k+2} = \Sigma(a_k/a_{k+1})$, which leads to

$$\frac{a_0}{a_n} = \frac{a_0}{a_1} \frac{a_1}{a_2} \cdots \frac{a_{n-1}}{a_n} = N_{\mathbb{Q}_q/\mathbb{Q}_p} \left(\frac{a_0}{a_1} \right).$$

If the curve $\overline{E}$ is defined by the equation $y^2 + xy = x^3 + \overline{c}$ with $\overline{c} \in \mathbb{F}_q^*$, then $(\sqrt[4]{\overline{c}}, \sqrt{\overline{c}})$ is a point of order 4, which implies that $\mathrm{Tr} \overline{F} \equiv 1 \pmod{4}$. Since $a_0/a_n \in 1 + 4\mathbb{Z}_q$, we can choose the correct sign in equation (VI.11) and conclude that $\mathrm{Tr} \overline{F} \equiv a_0/a_n \equiv N_{\mathbb{Q}_q/\mathbb{Q}_p}(\frac{a_0}{a_1}) \pmod{q}$.

VI.3.3. The AGM Algorithm. The only remaining problem is that we only have approximations to a and b and not the exact values as assumed in the previous section.

Let $\overline{E}$ be an ordinary elliptic curve defined by $y^2 + xy = x^3 + \overline{c}$ with $\overline{c} \in \mathbb{F}_q^*$. Take any $r \in \mathbb{Z}_q$ such that $r^2 \equiv \overline{c} \pmod{2}$ and let $a_0 = 1 + 4r$ and $b_0 = 1 - 4r$. In Section VI.3.1 we showed that if $(a_k, b_k)_{k=0}^\infty$ is the AGM sequence, then

$$j(E_{a_k, b_k}) \equiv j(\mathcal{E}_k) \pmod{2^{k+1}},$$

where $\mathcal{E}_k$ is the canonical lift of $\overline{\sigma}^k(\overline{E})$. Expressing $j(E_{a_k, b_k})$ as a function of a_k and b_k shows that a_k and b_k must be correct modulo 2^{k+3} . Therefore, we conclude that

$$\mathrm{Tr} \overline{F} \equiv \frac{a_{m-3}}{a_{m-3+n}} + 2^n \frac{a_{m-3+n}}{a_{m-3}} \pmod{2^m}.$$

This procedure is summarized in Algorithm VI.5.

ALGORITHM VI.5: AGM

INPUT: An elliptic curve $\overline{E}: y^2 + xy = x^3 + c$ over $\mathbb{F}_{2^n}$ with $j(\overline{E}) \notin \mathbb{F}_4$.

OUTPUT: The number of points on $\overline{E}(\mathbb{F}_{2^n})$.

1. $m \leftarrow \lceil \frac{n}{2} \rceil + 2$.
 2. **Take** $r \in \mathbb{Z}_q$ with $r \equiv \sqrt{c} \pmod{2}$.
 3. $a \leftarrow 1 + 4r$, $b \leftarrow 1 - 4r$.
 4. **For** $i = 4$ **to** m **do**:
 5. $(a, b) \leftarrow \left((a + b)/2, \sqrt{ab} \right) \pmod{2^i}$.
 6. $a_0 \leftarrow a$.
 7. **For** $i = 0$ **to** $n - 1$ **do**:
 8. $(a, b) \leftarrow \left((a + b)/2, \sqrt{ab} \right) \pmod{2^m}$.
 9. $t \leftarrow a_0/a \pmod{2^m}$.
 10. **If** $t^2 > 2^{n+2}$ **then** $t \leftarrow t - 2^m$.
 11. **Return** $2^n + 1 - t$.
-

The complexity of Algorithm VI.5 is determined by Step 7, which requires $O(n)$ square root computations to precision $m \simeq n/2$. Since each square root computation takes $O(1)$ multiplications at full precision, the complexity of Algorithm VI.5 is $O(n^{2\mu+1})$ bit-operations. The space complexity clearly is $O(n^2)$, since only $O(1)$ elements of $\mathbb{Z}_q/(2^m\mathbb{Z}_q)$ are required. Note that it is possible to replace the **For**-loop in Step 7 of Algorithm VI.5 by one AGM iteration and a norm computation. Indeed, the trace of Frobenius satisfies $t \equiv N_{\mathbb{Q}_q/\mathbb{Q}_p}(a_0/a_1) \pmod{2^{\min(n,m)}}$. We refer to Section VI.5 for efficient norm computation algorithms.

VI.3.4. Univariate AGM. Given the bivariate AGM sequence $(a_k, b_k)_{k=0}^\infty$ with $a_k \equiv b_k \equiv 1 \pmod{4}$ and $a_k \equiv b_k \pmod{8}$, we can easily define a univariate AGM sequence $(\lambda_k)_{k=0}^\infty$ by $\lambda_k = b_k/a_k$. This sequence corresponds to the elliptic curves

$$E_{\lambda_k}: y^2 = x(x-1)(x-\lambda_k^2).$$

Since $(a_{k+1}, b_{k+1}) = ((a_k + b_k)/2, \sqrt{a_k b_k})$, λ_{k+1} follows from λ_k by

$$\lambda_{k+1} = \frac{2\sqrt{\lambda_k}}{1 + \lambda_k}. \quad (\text{VI.12})$$

In Section VI.3.1 we showed that for an ordinary curve $\overline{E}: y^2 + xy = x^3 + \overline{c}$ with $\overline{c} \in \mathbb{F}_q^*$, the AGM sequence can be initialized as $(a_0, b_0) = (1 + 4u, 1 - 4u)$ where $u \in \mathbb{Z}_q$ satisfies $u^2 \equiv \overline{c} \pmod{2}$. However, these values cannot be used to initialize the univariate AGM sequence, since (a_0, b_0) is only correct modulo 8, which would lead to $\lambda_0 \equiv 1 \pmod{8}$. However, this problem can

be solved easily by computing $(a_1, b_1) \equiv (1, 1 - 8u^2) \pmod{16}$ and starting from

$$\lambda_1 \equiv 1 + 8u^2 \equiv 1 + 8c \pmod{16},$$

with $c \in \mathbb{Z}_q$ and $c \equiv \bar{c} \pmod{2}$. Unlike the bivariate AGM sequence, the univariate AGM sequence does converge, in the sense that $\lambda_k \equiv \lambda_{k+n} \pmod{2^{k+3}}$. In Section VI.3.2 we showed that

$$\frac{b_{k+1}}{a_{k+1}} \equiv \Sigma\left(\frac{b_k}{a_k}\right) \pmod{2^{k+3}},$$

and thus $\lambda_{k+1} \equiv \Sigma(\lambda_k) \pmod{2^{k+3}}$. Gaudry [143] suggested substituting this in equation (VI.12) which shows that λ_k satisfies

$$\Sigma(Z)^2(1+Z)^2 - 4Z \equiv 0 \pmod{2^{k+3}} \quad \text{and} \quad Z \equiv 1 + 8\Sigma^{k-1}(c) \pmod{16}.$$

Define the polynomial $\Lambda_2(X, Y) = Y^2(1+X)^2 - 4X$; then λ_k is a solution of $\Lambda_2(X, \Sigma(X)) \equiv 0 \pmod{2^{k+3}}$. Recall that the j -invariant of the canonical lift of $\overline{E}$ satisfies a similar type of equation, i.e., $\Phi_p(X, \Sigma(X)) = 0$ with $\Phi_p(X, Y)$ the p th modular polynomial. The main difference is that both partial derivatives of $\Lambda_2(X, Y)$ vanish modulo 2. Therefore, we make the change of variables $X \rightarrow 1 + 8X$ and $Y \rightarrow 1 + 8Y$ to obtain the modified modular polynomial

$$\Upsilon_2(X, Y) = (X + 2Y + 8XY)^2 + Y + 4XY.$$

Let γ_k be defined by $\lambda_k = 1 + 8\gamma_k$; then γ_k satisfies $\Upsilon_2(X, \Sigma(X)) \equiv 0 \pmod{2^k}$ and $\gamma_k \equiv \bar{\sigma}^{k-1}(c) \pmod{2}$. The partial derivatives of Υ_2 are

$$\begin{aligned} \frac{\partial \Upsilon_2}{\partial X}(X, Y) &= 2(X + 2Y + 8XY)(1 + 8Y) + 4Y, \\ \frac{\partial \Upsilon_2}{\partial Y}(X, Y) &= (4(X + 2Y + 8XY) + 1)(1 + 4X), \end{aligned}$$

which shows that $\frac{\partial \Upsilon_2}{\partial X} \equiv 0 \pmod{2}$, but $\frac{\partial \Upsilon_2}{\partial Y} \equiv 1 \pmod{2}$.

The trace of Frobenius follows easily from Section VI.3.2 and is given by

$$\text{Tr} \overline{F} \equiv N_{\mathbb{Q}_q/\mathbb{Q}_p} \left(\frac{1}{1 + 4\gamma_k} \right) \pmod{2^{\min(n, k)}}. \quad (\text{VI.13})$$

VI.4. Generalized Newton Iteration

In Sections VI.2.3 and VI.3.4 we solved equations of the form

$$\Gamma(X, \Sigma(X)) = 0 \text{ for } \Gamma(X, Y) \in \mathbb{Z}_q[X, Y]$$

without computing the Frobenius substitution Σ . The aim of this section is to show that in a well-chosen basis of $\mathbb{Q}_q$, the Frobenius substitution can be computed efficiently, which can be used to devise faster algorithms to solve the above equation $\Gamma(X, \Sigma(X)) = 0$.

VI.4.1. Fast Frobenius Substitution. The first construction admitting a fast Frobenius substitution is due to Satoh, Skjernaa and Taguchi [287] and works for all finite fields $\mathbb{F}_q \simeq \mathbb{F}_p[t]/(\bar{f}(t))$. Let $f(t) \in \mathbb{Z}_p[t]$ be the unique polynomial defined by

$$f(t)|(t^q - t) \quad \text{and} \quad f(t) \equiv \bar{f}(t) \pmod{p}.$$

Then $\mathbb{Q}_q$ can be represented as $\mathbb{Q}_p[t]/(f(t))$. The polynomial $f(t)$ can be computed modulo p^m in $O((nm)^\mu \log m)$ time using an algorithm due to Harley; a detailed description can be found in [333, 3.10.2].

As shown in Section VI.1, this implies that $\Sigma(t) = t^p$ and the Frobenius substitution of an element $\alpha = \sum_{i=0}^{n-1} \alpha_i t^i$ can be computed efficiently as

$$\Sigma(\alpha) = \Sigma \left(\sum_{i=0}^{n-1} \alpha_i t^i \right) = \sum_{i=0}^{n-1} \alpha_i t^{ip},$$

where the result is reduced modulo $f(t)$. Note that in general the polynomial $f(t)$ is dense, so a fast reduction algorithm should be used, e.g. the one described in [141, Chapter 9].

Similarly, the inverse Frobenius substitution Σ^{-1} can be computed as

$$\Sigma^{-1}(\alpha) = \Sigma^{-1} \left(\sum_{i=0}^{n-1} \alpha_i t^i \right) = \sum_{j=0}^{p-1} \left(\sum_{0 \leq pk+j < n} \alpha_{pk+j} t^k \right) C_j(t),$$

where $C_j(t) = \Sigma^{-1}(t^j) \equiv t^{jp^{n-1}} \pmod{f(t)}$. If we precompute $C_j(t)$ for $j = 0, \dots, p-1$, computing $\Sigma^{-1}(\alpha)$ for $\alpha \in \mathbb{Z}_q$ only takes $p-1$ multiplications in $\mathbb{Z}_q$.

The second construction is due to Kim, Park, Cheon, Park, Kim and Hahn [199] who proposed the use of finite fields with a Gaussian Normal Basis (GNB) of small type. Such a basis can be lifted trivially to $\mathbb{Z}_q$ and allows efficient computation of arbitrary iterates of Frobenius. Recall that a basis B of $\mathbb{Q}_q/\mathbb{Q}_p$ is called normal if there exists an element $\beta \in \mathbb{Q}_q$ such that $B = \{\Lambda(\beta) \mid \Lambda \in \text{Gal}(\mathbb{Q}_q/\mathbb{Q}_p)\}$. A proof of the next proposition can be found in [199].

PROPOSITION VI.12. *Let p be a prime and n, t positive integers such that $nt + 1$ is a prime different from p . Let γ be a primitive $(nt + 1)$ -th root of unity in some extension field of $\mathbb{Q}_p$. If $\gcd(nt/e, n) = 1$, with e the order of p modulo $nt + 1$, then for any primitive t th root of unity τ in $\mathbb{Z}/(nt + 1)\mathbb{Z}$*

$$\beta = \sum_{i=0}^{t-1} \gamma^{\tau^i}$$

is a normal element and $[\mathbb{Q}_p(\beta) : \mathbb{Q}_p] = n$. Such a basis is called a Gaussian Normal Basis of type t .

Kim et al. represent elements of $\mathbb{Z}_q$ as elements of the ring

$$\mathbb{Z}_p[x]/(x^{nt+1} - 1).$$

Multiplication of two elements in $\mathbb{Z}_q/(p^m\mathbb{Z}_q)$ therefore requires $O((nmt)^\mu)$ bit-operations, so one normally restricts $t \leq 2$.

For $t = 1$ we have $\beta = \gamma$ and the minimal polynomial of β therefore is

$$f(x) = \frac{x^{n+1} - 1}{x - 1} = x^n + x^{n-1} + \cdots + x + 1.$$

To speed up the Frobenius substitution, Kim et al. use a redundant representation, i.e., they embed $\mathbb{Z}_q$ in $\mathbb{Z}_p[x]/(x^{n+1} - 1)$ by mapping $\alpha = \sum_{i=0}^{n-1} \alpha_i \beta^i$ to $\alpha(x) = \sum_{i=0}^{n-1} \alpha_i x^i + 0x^n$. Since $\Sigma^k(\beta) = \beta^{p^k}$, we have

$$\Sigma^k(\alpha(x)) = \sum_{i=0}^n \alpha_i x^{ip^k} = a_0 + \sum_{j=1}^n \alpha_{j/p^k \pmod{(n+1)}} x^j.$$

Therefore, we can compute $\Sigma^k(\alpha)$ by a simple permutation of the coefficients of $\alpha(x)$, which only requires $O(n)$ bit-operations.

VI.4.2. Satoh–Skjernaa–Taguchi Algorithm. Let $x \in \mathbb{Z}_q$ be a root of $\Gamma(X, \Sigma(X)) = 0$ for $\Gamma(X, Y) \in \mathbb{Z}_q[X, Y]$ and assume we have computed the approximation $x_m \equiv x \pmod{p^m}$. Define $\delta_m = (x - x_m)/p^m$; then the Taylor series expansion around x_m gives

$$\begin{aligned} 0 &= \Gamma(x, \Sigma(x)) = \Gamma(x_m + p^m \delta_m, \Sigma(x_m + p^m \delta_m)) \\ &\equiv \Gamma(x_m, \Sigma(x_m)) + p^m (\delta_m \Delta_x + \Sigma(\delta_m) \Delta_y) \pmod{p^{2m}}, \end{aligned} \tag{VI.14}$$

with $\Delta_x \equiv \frac{\partial \Gamma}{\partial X}(x_m, \Sigma(x_m)) \pmod{p^m}$ and $\Delta_y \equiv \frac{\partial \Gamma}{\partial Y}(x_m, \Sigma(x_m)) \pmod{p^m}$. Since $\Gamma(x_m, \Sigma(x_m)) \equiv 0 \pmod{p^m}$, we can divide this equation by p^m and obtain a relation for δ_m modulo p^m :

$$\frac{\Gamma(x_m, \Sigma(x_m))}{p^m} + \delta_m \Delta_x + \Sigma(\delta_m) \Delta_y \equiv 0 \pmod{p^m}.$$

For simplicity we will assume that $\text{ord}_p(\Delta_y) = 0$, i.e., that Δ_y is a unit in $\mathbb{Z}_q$ and that $\text{ord}_p(\Delta_x) > 0$. Reducing the above equation modulo p then gives

$$\delta_m^p \equiv -\frac{\Gamma(x_m, \Sigma(x_m))}{p^m \Delta_y} \pmod{p}.$$

Taking the unique p th root $\delta_m \in \mathbb{F}_q$ leads to a better approximation of x given by $x_m + p^m \delta_m \equiv x \pmod{p^{m+1}}$. To avoid computing the p th root in the above equation, Satoh, Skjernaa and Taguchi replace the equation $\Gamma(X, \Sigma(X)) = 0$ by $\Gamma(\Sigma^{-1}(X), X) = 0$, which clearly is equivalent. The new δ_m then becomes

$$\delta_m \equiv -\frac{\Gamma(\Sigma^{-1}(x_m), x_m)}{p^m \frac{\partial \Gamma}{\partial Y}(\Sigma^{-1}(x_m), x_m)} \pmod{p}.$$

Note that since $\Gamma(\Sigma^{-1}(x_m), x_m) \equiv 0 \pmod{p^m}$, we only need to compute the inverse of $\frac{\partial \Gamma}{\partial Y}(\Sigma^{-1}(x_m), x_m)$ modulo p . Implementing these ideas naively leads to Algorithm VI.6, which can be used instead of Algorithm VI.4.

ALGORITHM VI.6: Satoh_Skjernaa_Taguchi_Naive

INPUT: Polynomial $\Gamma(X, Y) \in \mathbb{Z}_q$, element $x_0 \in \mathbb{Z}_q$ satisfying $\Gamma(\Sigma^{-1}(x_0), x_0) \equiv 0 \pmod{p}$ and precision m .
OUTPUT: Element $x_m \in \mathbb{Z}_q$, with $\Gamma(\Sigma^{-1}(x_m), x_m) \equiv 0 \pmod{p^m}$ and $x_m \equiv x_0 \pmod{p}$.

1. $d \leftarrow \left(\frac{\partial \Gamma}{\partial Y}(\Sigma^{-1}(x_0), x_0) \right)^{-1} \pmod{p}$.
2. $y \leftarrow x_0 \pmod{p}$.
3. **For** $i = 2$ **to** m **do**:
4. $x \leftarrow \Sigma^{-1}(y) \pmod{p^i}$.
5. $y \leftarrow y - d\Gamma(x, y) \pmod{p^i}$.
6. **Return** y .

The main problem of Algorithm VI.6 is that in each iteration it recomputes $\Gamma(x, y)$ although the values of x and y in Step $i + 1$ are very close to the ones in Step i . Assume we have computed $x_W \equiv x \pmod{p^W}$ for some W . Then, for all $s \in \mathbb{N}$,

$$\Gamma(\Sigma^{-1}(x_{sW+i}), x_{sW+i}) \equiv \Gamma(\Sigma^{-1}(x_{sW}), x_{sW}) + \Delta \pmod{p^{(s+1)W}}, \quad (\text{VI.15})$$

with

$$\Delta = p^{sW} \left(\frac{\partial \Gamma}{\partial X}(\Sigma^{-1}(x_{sW}), x_{sW}) \Sigma^{-1}(\delta) + \frac{\partial \Gamma}{\partial Y}(\Sigma^{-1}(x_{sW}), x_{sW}) \delta \right).$$

Furthermore, note that it is sufficient to know the partial derivatives

$$\frac{\partial \Gamma}{\partial X}(\Sigma^{-1}(x_{sW}), x_{sW}) \quad \text{and} \quad \frac{\partial \Gamma}{\partial Y}(\Sigma^{-1}(x_{sW}), x_{sW})$$

modulo p^W only. Using (VI.15), we can thus compute $\Gamma(\Sigma^{-1}(x_{sW+i}), x_{sW+i})$ from $\Gamma(\Sigma^{-1}(x_{sW}), x_{sW})$ as long as $i < W$.

Algorithm VI.7 consists of two stages: in the first stage we compute $x_W \equiv x \pmod{p^W}$ using Algorithm VI.6 and in the second stage we use equation (VI.15) to update $\Gamma(x, y)$.

ALGORITHM VI.7: Satoh_Skjernaa_Taguchi

INPUT: Polynomial $\Gamma(X, Y) \in \mathbb{Z}_q$, element $x_0 \in \mathbb{Z}_q$ satisfying $\Gamma(\Sigma^{-1}(x_0), x_0) \equiv 0 \pmod{p}$ and precision m .

OUTPUT: Element $x_m \in \mathbb{Z}_q$, with $\Gamma(\Sigma^{-1}(x_m), x_m) \equiv 0 \pmod{p^m}$ and $x_m \equiv x_0 \pmod{p}$.

1. $y \leftarrow \text{Satoh_Skjernaa_Taguchi_Naive}(x_0, W)$.
2. $x \leftarrow \Sigma^{-1}(y) \pmod{p^W}$.
3. $\Delta_x \leftarrow \frac{\partial \Gamma}{\partial X}(x, y) \pmod{p^W}$.
4. $\Delta_y \leftarrow \frac{\partial \Gamma}{\partial Y}(x, y) \pmod{p^W}$.
5. For $s = 1$ to $\lfloor (m-1)/W \rfloor$ do:
6. $x \leftarrow \Sigma^{-1}(y) \pmod{p^{(s+1)W}}$.
7. $V \leftarrow \Gamma(x, y) \pmod{p^{(s+1)W}}$.
8. For $i = 0$ to $W-1$ do:
9. $\delta_y \leftarrow -dp^{-(sW+i)}V \pmod{p}$.
10. $\delta_x \leftarrow \Sigma^{-1}(\delta_y) \pmod{p^{W-i}}$.
11. $y \leftarrow y + p^{sW+i}\delta_y \pmod{p^{(s+1)W}}$.
12. $V \leftarrow V + p^{(sW+i)}(\Delta_x\delta_x + \Delta_y\delta_y) \pmod{p^{(s+1)W}}$.
13. Return y .

Satoh, Skjernaa and Taguchi prove that for $W \simeq n^{\mu/(1+\mu)}$, Algorithm VI.7 runs in time $O(n^\mu m^{\mu+1/(1+\mu)})$. Although the value for W minimizes the time complexity, in practise it is faster to take W a multiple of the CPU word size.

VI.4.3. Solving Artin–Schreier Equations. Recall that if $\mathbb{F}_q$ is a field of characteristic p , an equation of the form $x^p - x + \bar{c} = 0$ with $\bar{c} \in \mathbb{F}_q$ is called an Artin–Schreier equation. Since $\Sigma(x) \equiv x^p \pmod{p}$, a natural generalization to $\mathbb{Z}_q$ is an equation of the form

$$a\Sigma(x) + bx + c = 0, \quad (\text{VI.16})$$

with $a, b, c \in \mathbb{Z}_q$. Let $x \in \mathbb{Z}_q$ be a solution of $\Gamma(X, \Sigma(X)) = 0$ and let $x_m \equiv x \pmod{p^m}$. Define $\delta_m = (x - x_m)/p^m$; then equation (VI.14) shows that δ_m satisfies an Artin–Schreier equation modulo p^m with

$$a = \frac{\partial \Gamma}{\partial Y}(x_m, \Sigma(x_m)), \quad b = \frac{\partial \Gamma}{\partial X}(x_m, \Sigma(x_m)), \quad c = \frac{\Gamma(x_m, \Sigma(x_m))}{p^m}.$$

Since we assumed that a is a unit in $\mathbb{Z}_q$, any solution δ to this Artin–Schreier equation satisfies $\delta \equiv \delta_m \pmod{p^m}$, and thus $x_m + p^m\delta \equiv x \pmod{p^{2m}}$. Therefore, if we can solve Artin–Schreier equations efficiently, we obtain an algorithm to find a root of $\Gamma(X, \Sigma(X)) = 0$ that converges quadratically. This strategy is formalized in Algorithm VI.8. Since the precision we compute with doubles in every step, the complexity of Algorithm VI.8 is the same as the complexity of the algorithm to solve an Artin–Schreier equation.

ALGORITHM VI.8: Gen_Newton_Lift

INPUT: Polynomial $\Gamma(X, Y) \in \mathbb{Z}_q$, element $x_0 \in \mathbb{Z}_q$ satisfying $\Gamma(x_0, \Sigma(x_0)) \equiv 0 \pmod{p}$ and precision m .
OUTPUT: Element $x_m \in \mathbb{Z}_q$, with $\Gamma(x_m, \Sigma(x_m)) \equiv 0 \pmod{p^m}$ and $x_m \equiv x_0 \pmod{p}$.

1. If $m = 1$ then
2. $x \leftarrow x_0$.
3. Else
4. $m' \leftarrow \lceil \frac{m}{2} \rceil$.
5. $x' \leftarrow \text{Gen_Newton_Lift}(\Gamma, x_0, m')$;
6. $y' \leftarrow \Sigma(x') \pmod{p^{m'}}$.
7. $V \leftarrow \Gamma(x', y') \pmod{p^{m'}}$.
8. $\Delta_x \leftarrow \frac{\partial \Gamma}{\partial X}(x', y') \pmod{p^{m'}}$.
9. $\Delta_y \leftarrow \frac{\partial \Gamma}{\partial Y}(x', y') \pmod{p^{m'}}$.
10. $\delta \leftarrow \text{Artin_Schreier_Root}(\Delta_y, \Delta_x, V/p^{m'}, m - m')$.
11. $x \leftarrow x' + p^{m'} \delta \pmod{p^m}$.
12. Return x .

Lercier–Lubicz Algorithm: In [218] Lercier and Lubicz described the first quadratically convergent algorithm to solve Artin–Schreier equations. Since the algorithm uses iterated Frobenius substitutions, it is only efficient if $\mathbb{Q}_q$ admits a Gaussian Normal Basis of small type.

Dividing equation (VI.16) by a , we obtain

$$\Sigma(x) - \beta x - \gamma = 0,$$

with $\beta = -b/a, \gamma = -c/a \in \mathbb{Z}_q$. Let $\beta_1 = \beta, \gamma_1 = \gamma$; then $\Sigma(x) = \beta_1 x + \gamma_1$ and define $\Sigma^k(x) = \beta_k x + \gamma_k$ for $k = 2, \dots, n$ by induction. Since $\Sigma^n(x) = x$ for all $x \in \mathbb{Q}_q$, we conclude that the unique solution to the above equation is given by $\gamma_n/(1 - \beta_n)$. To compute the $\beta_k, \gamma_k \in \mathbb{Z}_q$, Lercier and Lubicz use a simple square and multiply algorithm based on the formula

$$\Sigma^{k+l}(x) = \Sigma^l(\beta_k x + \gamma_k) = \Sigma^l(\beta_k)(\beta_l x + \gamma_l) + \Sigma^l(\gamma_k).$$

The above formula immediately leads to Algorithm VI.9, which returns β_k and γ_k modulo p^m for any k .

The complexity of Algorithm VI.9 is determined by Steps 6 and 7, which require $O(1)$ multiplications and $O(1)$ iterated Frobenius substitutions in $\mathbb{Z}_q/(p^m \mathbb{Z}_q)$. For fields with a Gaussian Normal Basis of small type, such a substitution takes $O(n)$ bit-operations as shown in Section VI.4.1 and a multiplication takes $O((nm)^\mu)$ bit-operations. Since the algorithm needs $O(\log n)$ recursive calls, the time and space complexity for fields with Gaussian Normal Basis of small type are $O((nm)^\mu \log n)$ and $O(nm)$, respectively.

ALGORITHM VI.9: Lercier-Lubicz

INPUT: Elements $\beta, \gamma \in \mathbb{Z}_q$, power k , precision m .

OUTPUT: Elements $\beta_k, \gamma_k \in \mathbb{Z}_q$ such that $\Sigma^k(x) \equiv \beta_k x + \gamma_k \pmod{p^m}$
and $\Sigma(x) \equiv \beta x + \gamma \pmod{p^m}$.

1. If $k = 1$ then
 2. $\beta_k \leftarrow \beta \pmod{p^m}, \gamma_k \leftarrow \gamma \pmod{p^m}$.
 3. Else
 4. $k' \leftarrow \lfloor \frac{k}{2} \rfloor$.
 5. $(\beta_{k'}, \gamma_{k'}) \leftarrow \text{Lercier-Lubicz}(\beta, \gamma, k', m)$.
 6. $\beta_k \leftarrow \beta_{k'} \Sigma^{k'}(\beta_{k'}) \pmod{p^m}$.
 7. $\gamma_k \leftarrow \gamma_{k'} \Sigma^{k'}(\beta_{k'}) + \Sigma^{k'}(\gamma_{k'}) \pmod{p^m}$.
 8. If $k \equiv 1 \pmod{2}$ then
 9. $\beta_k \leftarrow \beta \Sigma(\beta_k) \pmod{p^m}$.
 10. $\gamma_k \leftarrow \gamma \Sigma(\beta_k) + \Sigma(\gamma_k) \pmod{p^m}$.
 11. Return (β_k, γ_k) .
-

Harley's Algorithm: In an e-mail to the NMBRTHRY list [158], Harley announced an algorithm to count the number of points on an elliptic curve over $\mathbb{F}_{p^n}$ which runs in $O(n^{2\mu} \log n)$ time and uses $O(n^2)$ space for fixed p . Note that this algorithm is protected by a U.S. patent [159] filed by Harley. The key idea is a doubly recursive algorithm to solve the Artin-Schreier equation

$$a \Sigma(x) + bx + c \equiv 0 \pmod{p^m}, \quad (\text{VI.17})$$

with $a, b, c \in \mathbb{Z}_q$, $\text{ord}_p(a) = 0$ and $\text{ord}_p(b) > 0$. Let $m' = \lceil m/2 \rceil$ and assume we have an algorithm which returns a zero $x_{m'}$ of (VI.17) to precision m' . Then we can use the same algorithm to find a solution x_m to precision m . Indeed, substituting $x_m = x_{m'} + p^{m'} \Delta_m$ in equation (VI.17) and dividing by $p^{m'}$ gives

$$a \Sigma(\Delta_m) + b \Delta_m + \frac{a \Sigma(x_{m'}) + b x_{m'} + c}{p^{m'}} \equiv 0 \pmod{p^{m-m'}}.$$

This shows that Δ_m itself also satisfies an Artin-Schreier equation of the form (VI.17) but to a lower precision. Since $m - m' \leq m'$ we can use the same algorithm to determine $\Delta_m \pmod{p^{m-m'}}$ and therefore x_m .

This immediately leads to a doubly recursive algorithm if we can solve the base case, i.e., find a solution to (VI.17) modulo p . Since we assumed that $\text{ord}_p(a) = 0$ and $\text{ord}_p(b) > 0$, we simply have to solve $ax^p + c \equiv 0 \pmod{p}$.

ALGORITHM VI.10: Harley

INPUT: Elements $a, b, c \in \mathbb{Z}_q$, $a \in \mathbb{Z}_q^\times$, $\text{ord}_p(b) > 0$, precision m .
OUTPUT: Element $x \in \mathbb{Z}_q$ such that $a\Sigma(x) + bx + c \equiv (\text{mod } p^m)$.

1. If $m = 1$ then
 2. $x \leftarrow (-c/a)^{1/p} \pmod{p}$.
 3. Else
 4. $m' \leftarrow \lceil \frac{m}{2} \rceil$; $M = m - m'$.
 5. $x' \leftarrow \text{Harley}(a, b, c, m')$.
 6. $c' \leftarrow \frac{a\Sigma(x') + bx' + c}{p^{m'}} \pmod{p^M}$.
 7. $\Delta' \leftarrow \text{Harley}(a, b, c', M)$.
 8. $x \leftarrow x' + p^{m'}\Delta' \pmod{p^m}$.
 9. Return x .
-

The p th root in Step 2 of Algorithm VI.10 should not be computed by naively taking the p^{n-1} th power. Instead, let $\mathbb{F}_q = \mathbb{F}_p[\bar{\theta}]$; then

$$\left(\sum_{i=0}^{n-1} \bar{a}_i \bar{\theta}^i \right)^{1/p} = \sum_{j=0}^{p-1} \left(\sum_{0 \leq pk+j < n} \bar{a}_{pk+j} \bar{\theta}^k \right) C_j(\bar{\theta}),$$

with $C_j(\bar{\theta}) = (\bar{\theta}^j)^{1/p} = \bar{\theta}^{jp^{n-1}}$. This shows that for $\bar{z} \in \mathbb{F}_q$, we can compute $\bar{z}^{1/p}$ with $p-1$ multiplications over $\mathbb{F}_q$.

The complexity of Algorithm VI.10 is determined by the recursive calls in Steps 5 and 7 and the Frobenius substitution in Step 6.

If we assume that $\mathbb{Z}_q$ is represented as in Section VI.4.1, the Frobenius substitution in $\mathbb{Z}_q/(p^m\mathbb{Z}_q)$ can be computed using $O((nm)^\mu)$ bit-operations. If $T(m)$ is the running time of Algorithm VI.10 for precision m , then we have

$$T(m) \leq 2T(\lceil m/2 \rceil) + c(nm)^\mu,$$

for some constant c . The above relation implies by induction that the time complexity of Algorithm VI.10 is $O((nm)^\mu \log m)$. The space complexity is clearly given by $O(nm)$.

VI.5. Norm Computation

The final step in all point counting algorithms is to compute the norm $N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) = \prod_{i=0}^{n-1} \Sigma^i(\alpha)$ of a unit $\alpha \in \mathbb{Z}_q$. In this section we give an overview of the existing algorithms to compute $N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha)$.

VI.5.1. Norm Computation I. Kedlaya [196] suggested a basic square and multiply approach by computing

$$\alpha_{i+1} = \Sigma^{2^i}(\alpha_i) \alpha_i \quad \text{for } i = 0, \dots, \lfloor \log_2 n \rfloor,$$

with $\alpha_0 = \alpha$, and to combine these to recover $N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) = \Sigma^{n-1}(\alpha) \cdots \Sigma(\alpha) \alpha$. Let $n = \sum_{i=0}^l n_i 2^i$, with $n_i \in \{0, 1\}$, and $n_l = 1$; then we can write

$$N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) = \prod_{i=0}^l \Sigma^{2^{i+1}+2^{i+2}+\dots+2^l}(\alpha_i^{n_i}),$$

where the sum $2^{i+1} + \dots + 2^l$ is defined zero for $i \geq l$. This formula immediately leads to Algorithm VI.11.

ALGORITHM VI.11: Norm_I

INPUT: An element $\alpha \in \mathbb{Z}_q$ with $q = p^n$ and a precision m .

OUTPUT: The norm $N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) \pmod{p^m}$.

1. $i \leftarrow n$, $j \leftarrow 0$, $r \leftarrow 1$, $s \leftarrow \alpha$.
 2. **While** $i > 0$ **do**:
 3. **If** $i \equiv 1 \pmod{2}$ **then** $r \leftarrow \Sigma^{2^j}(r) s \pmod{p^m}$.
 4. **If** $i > 1$ **then** $s \leftarrow \Sigma^{2^j}(s) s \pmod{p^m}$.
 5. $j \leftarrow j + 1$, $i \leftarrow \lfloor i/2 \rfloor$.
 6. **Return** r .
-

This algorithm is particularly attractive for p -adic fields with Gaussian Normal Basis of small type, due to efficient repeated Frobenius substitutions. In this case the time complexity is determined by the $O(\log n)$ multiplications in $\mathbb{Z}_q/(p^m \mathbb{Z}_q)$, which amount to $O((nm)^\mu \log n)$ bit-operations and the space complexity is $O(nm)$.

VI.5.2. Norm Computation II. Satoh, Skjernaa and Taguchi [287] also proposed a fast norm computation algorithm based on an analytic method. Let $\exp(x) = \sum_{i=0}^{\infty} x^i/i!$ and $\log(x) = \sum_{i=1}^{\infty} (-1)^{i-1}(x-1)^i/i$ be the p -adic exponential and logarithmic function. An easy calculation shows that $\exp(x)$ converges for $\text{ord}_p(x) > 1/(p-1)$ and $\log(x)$ converges for $\text{ord}_p(x-1) > 0$.

Assume first that α is close to unity, i.e., $\text{ord}_p(\alpha - 1) > 1/(p-1)$; then

$$N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) = \exp \left(\text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\log(\alpha)) \right),$$

since Σ is continuous and both series converge. The dominant step in the above formula is the evaluation of the logarithm. Using a naive algorithm, this would take $O(m)$ multiplications over $\mathbb{Z}_q/(p^m \mathbb{Z}_q)$ or $O(n^\mu m^{\mu+1})$ bit-operations.

Satoh, Skjernaa and Taguchi solve this problem by noting that α^{p^k} for $k \in \mathbb{N}$ is very close to unity, i.e., $\text{ord}_p(\alpha^{p^k} - 1) > k + \frac{1}{p-1}$. If $\alpha \in \mathbb{Z}_q/(p^m \mathbb{Z}_q)$,

then α^{p^k} is well defined in $\mathbb{Z}_q/(p^{m+k}\mathbb{Z}_q)$ and can be computed with $O(k)$ multiplications in $\mathbb{Z}_q/(p^{m+k}\mathbb{Z}_q)$. Furthermore, note that

$$\log(\alpha) \equiv p^{-k}(\log(\alpha^{p^k}) \pmod{p^{m+k}}) \pmod{p^m}$$

and that $\log(a^{p^k}) \pmod{p^{m+k}}$ can be computed with $O(m/k)$ multiplications over $\mathbb{Z}_q/(p^{m+k}\mathbb{Z}_q)$. So if we take $k \simeq \sqrt{m}$, then $\log(\alpha) \pmod{p^m}$ can be computed in $O(n^\mu m^{\mu+0.5})$ bit-operations.

In characteristic 2, we can use an extra trick. Since $\text{ord}_2(\alpha - 1) > 1$, we have $\alpha \equiv 1 \pmod{2^\nu}$ for $\nu \geq 2$. Let $z = \alpha - 1 \in 2^\nu \mathbb{Z}_q/(2^m \mathbb{Z}_q)$ and define $\gamma = \frac{z}{2+z} \in 2^{\nu-1} \mathbb{Z}_q/(2^{m-1} \mathbb{Z}_q)$. Then $\alpha = 1 + z = \frac{1+\gamma}{1-\gamma}$ and thus

$$\log(\alpha) = \log(1 + z) = 2 \sum_{j=1}^{\infty} \frac{\gamma^{2j-1}}{2j-1}.$$

Note that all the denominators in the above formula are odd. Reducing this equation modulo 2^m therefore leads to

$$\log(\alpha) \equiv \log(1 + z) \equiv 2 \sum_{1 \leq (\nu-1)(2j-1) < m-1} \frac{\gamma^{2j-1}}{2j-1} \pmod{2^m}.$$

To compute the trace $\text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}$, Satoh, Skjernaa and Taguchi proceed as follows: define $\alpha_i \in \mathbb{Z}_p$ by $\log(\alpha) \equiv \sum_{i=0}^{n-1} \alpha_i t^i \pmod{p^m}$; then

$$\text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\log(\alpha)) \equiv \sum_{i=0}^{n-1} \alpha_i \text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(t^i) \pmod{p^m}$$

and each $\text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(t^i)$ for $i = 0, \dots, n-1$ can be precomputed using Newton's formula:

$$\text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(t^i) + \sum_{j=1}^{i-1} \text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(t^{i-j}) f_{n-j} + i f_{n-i} \equiv 0 \pmod{p^m},$$

with $f(t) = \sum_{i=0}^n f_i t^i$ the defining polynomial of $\mathbb{Q}_q \cong \mathbb{Q}_p[t]/(f(t))$.

Let $t_a = \text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(\log(a)) \in \mathbb{Z}_p$. Then $\text{ord}_p(t_a) > 1/(p-1)$, so $t_a \in p\mathbb{Z}_p$ for $p \geq 3$ and $t_a \in 4\mathbb{Z}_2$ for $p = 2$. If we precompute $\exp(p) \pmod{p^m}$ for $p \geq 3$ or $\exp(4) \pmod{2^m}$ for $p = 2$, then

$$\exp(t_a) \equiv \exp(p)^{t_a/p} \pmod{p^m} \quad \text{for } p \geq 3,$$

$$\exp(t_a) \equiv \exp(4)^{t_a/4} \pmod{2^m} \quad \text{for } p = 2,$$

and we can use a simple square and multiply algorithm to compute $\exp(t_a)$. Therefore, if α is close to unity, then $N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) \pmod{p^m}$ can be computed in $O(n^\mu m^{\mu+0.5})$ bit-operations and $O(nm)$ space. Algorithm VI.12 computes the norm of an element in $1 + 2^\nu \mathbb{Z}_q$, with $q = 2^n$, assuming that $\exp(4)$ and $\text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(t^i)$ for $i = 0, \dots, n-1$ are precomputed.

ALGORITHM VI.12: Norm_II

INPUT: An element $\alpha \in 1 + 2^\nu \mathbb{Z}_q$ with $\nu \geq 2$ and a precision m .

OUTPUT: The norm $N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) \pmod{2^m}$.

1. $s \leftarrow \lfloor \sqrt{m}/2 \rfloor$.
 2. $z \leftarrow \alpha^{2^s} - 1 \pmod{2^{m+s}}$.
 3. $w \leftarrow \log(1 + z) \pmod{2^{m+s}}$.
 4. $w \leftarrow 2^{-s}w \pmod{2^m}$.
 5. $u \leftarrow 2^{-\nu} \text{Tr}_{\mathbb{Q}_q/\mathbb{Q}_p}(w) \pmod{2^{m-\nu}}$.
 6. **Return** $\exp(4)^u \pmod{2^m}$.
-

Equations (VI.5) and (VI.13) show that for $p = 2$ we can indeed assume that $\text{ord}_p(\alpha - 1) > 1/(p - 1)$; however, for $p \geq 3$ we also need to consider the more general situation where $\alpha \in \mathbb{Z}_q$ is not close to unity. Let $\bar{\alpha} \in \mathbb{F}_q$ denote the reduction of α modulo p and $\alpha_t \in \mathbb{Z}_q$ the Teichmüller lift of $\bar{\alpha}$, i.e., the unique $(q - 1)$ th root of unity which reduces to $\bar{\alpha}$. Consider the equality

$$N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) = N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha_t) N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha_t^{-1} \alpha);$$

then $\text{ord}_p(\alpha_t^{-1} \alpha - 1) \geq 1 > 1/(p - 1)$ since p is odd. Furthermore, note that $N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha_t)$ is equal to the Teichmüller lift of $N_{\mathbb{F}_q/\mathbb{F}_p}(\bar{\alpha})$. Satoh [286] pointed out that the Teichmüller lift α_t can be computed efficiently as the solution of

$$\Sigma(X) = X^p \quad \text{and} \quad X \equiv \bar{\alpha} \pmod{p}.$$

Note that we can apply the same algorithms as in Section VI.4.

VI.5.3. Norm Computation III. In an e-mail to the NMBRTHRY list [158], Harley suggested an asymptotically fast norm computation algorithm based on a formula from number theory that expresses the norm as a resultant. The resultant itself can be computed using an adaptation of Moenck's fast extended GCD algorithm [252].

Let $\mathbb{Z}_q \cong \mathbb{Z}_p[t]/(f(t))$ with $f(t) \in \mathbb{Z}_p[t]$ a monic irreducible polynomial of degree n . Let $\theta \in \mathbb{Z}_q$ be a root of $f(t)$; then $f(t)$ splits completely over $\mathbb{Z}_q$ as $f(t) = \prod_{i=0}^{n-1} (x - \Sigma^i(\theta))$. For $\alpha = \sum_{i=0}^{n-1} \alpha_i \theta^i \in \mathbb{Z}_q$, define the polynomial $A(x) = \sum_{i=0}^{n-1} \alpha_i x^i \in \mathbb{Z}_p[x]$. By definition of the norm and the resultant we have

$$N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) = \prod_{i=0}^{n-1} \Sigma^i(\alpha) = \prod_{i=0}^{n-1} A(\Sigma^i(\theta)) = \text{Res}(f(x), A(x)).$$

The resultant $\text{Res}(f(x), A(x))$ can be computed in softly linear time using a variant of Moenck's fast extended GCD algorithm [252] as described in detail in Chapter 11 of the book by von zur Gathen and Gerhard [141]. The result is an algorithm to compute $N_{\mathbb{Q}_q/\mathbb{Q}_p}(\alpha) \pmod{p^m}$ in time $O((nm)^\mu \log n)$.

VI.6. Concluding Remarks

The p -adic approach, introduced by Satoh at the end of 1999, has caused a fundamental dichotomy in point counting algorithms that essentially depends on the characteristic of the finite field. For elliptic curves over finite fields $\mathbb{F}_{p^n}$ with p moderately large, the l -adic approach, i.e., the SEA algorithm, remains the only efficient solution and requires $O(n^{2\mu+2} \log^{2\mu+2} p)$ time. This stands in stark contrast with the p -adic approach which only requires $O(n^{2\mu} \log n)$ time for fixed p . The O -constant grows exponentially in p though, which implies that p -adic algorithms are only efficient for small p . The huge efficiency gap between the l -adic and p -adic approaches in small characteristic is illustrated by the following examples: for an elliptic curve over $\mathbb{F}_{2^{163}}$ (resp. $\mathbb{F}_{2^{509}}$) the SEA algorithm runs about two (resp. three) orders of magnitude slower than the p -adic algorithms.

In this chapter we have only considered the advances in elliptic curve point counting algorithms. To provide a starting point, we briefly review the most important developments for higher genus curves, especially hyperelliptic curves. The p -adic approach currently comes in two flavours:

Canonical Lift / AGM. This method computes the Serre-Tate canonical lift of an ordinary abelian variety and recovers the zeta function from the action of Frobenius on a basis for the holomorphic differentials. The main references are as follows: Mestre [247, 248] generalized the AGM algorithm to ordinary hyperelliptic curves over $\mathbb{F}_{2^n}$. An optimized version by Lercier and Lubicz [219] runs in time $O(2^g n^{2\mu})$ for an ordinary genus g hyperelliptic curve over $\mathbb{F}_{2^n}$. Ritzenthaler [282] extended Mestre's algorithm to ordinary genus 3 non-hyperelliptic curves.

p -adic Cohomology. These methods compute the action of a Frobenius operator on p -adic cohomology groups. Different cohomology theories give rise to different algorithms: Lauder and Wan [213] used p -adic analysis à la Dwork [113] to show that the zeta function of any algebraic variety can be computed in polynomial time. Lauder and Wan [214, 215] also devised an algorithm based on Dwork cohomology to compute the zeta function of an Artin-Schreier curve in time $O(g^{2\mu+3} n^{3\mu})$. Kedlaya [196] gave algorithms to compute with Monsky-Washnitzer cohomology and obtained an $O(g^{2\mu+2} n^{3\mu})$ time algorithm to compute the zeta function of a hyperelliptic curve in odd characteristic. Kedlaya's algorithm was subsequently extended as follows: to characteristic 2 by Denef and Vercauteren [96, 97]; to superelliptic curves by Gaudry and Gürel [144] and, finally, to C_{ab} curves by Denef and Vercauteren [98]. A nice introduction to these p -adic cohomology theories can be found in the PhD thesis of Gerkmann [148].

CHAPTER VII

Hyperelliptic Curves and the HCDLP

P. Gaudry

In 1989, hyperelliptic curves were proposed by Koblitz for use in cryptography as an alternative to elliptic curves. Recent improvements on the algorithms for computing the group law tend to prove that indeed cryptosystems based on hyperelliptic curves can be competitive. However, another interest of the study of hyperelliptic curves – and also of more general curves – is that the process called the Weil descent relates some instances of the ECDLP to instances of the hyperelliptic discrete logarithm problem (HCDLP). The purpose of this chapter is to give efficient algorithms for hyperelliptic curves that are useful for the next chapter.

VII.1. Generalities on Hyperelliptic Curves

In this section, we recall briefly the basic definitions and properties of hyperelliptic curves and their Jacobians. A complete elementary introduction to hyperelliptic curves that includes all the proofs can be found in [243]. For a more general treatment see [226].

VII.1.1. Equation of a Hyperelliptic Curve. Hyperelliptic curves can be defined in an abstract manner as follows: a hyperelliptic curve is a curve which is a covering of degree 2 of the projective line. In a more concrete setting, a hyperelliptic curve $\mathcal{C}$ of genus g is a curve with an equation of the form

$$\mathcal{C} : Y^2 + H(X)Y = F(X),$$

where F and H are polynomials with $\deg F = 2g+1$ and $\deg H \leq g$. Furthermore we ask the curve to be non-singular as an affine curve, which translates into the non-simultaneous vanishing of the partial derivatives of the equation.

This choice of equation is not always possible. In general, the polynomial $H(X)$ has degree at most $g+1$ and $F(X)$ has degree at most $2g+2$. The condition for being able to reduce the degrees to g and $2g+1$ is to have a rational Weierstraß point, which is equivalent to saying that $H(X)$ has a rational root if the characteristic is two, or that $4F(X) + H(X)^2$ has a rational root if the characteristic is odd. For this, we apply an invertible transformation $X' = \frac{aX+b}{cX+d}$ which sends this rational root to infinity. If the polynomials F and H have degrees $2g+1$ and at most g , we call the corresponding equation an *imaginary* model of the curve. Otherwise, the equation is said to be a *real*

model of the curve. In the following, we always consider an imaginary model. Using the real model introduces some extra technicalities, since there are now zero or two points at infinity.

In the case where the characteristic is odd, it is also possible to set $H(X) = 0$ without loss of generality. Indeed, we can apply the transformation $Y' = Y + H(X)/2$, which kills the crossed term. The division by two, prevents one from doing the same in characteristic two.

A hyperelliptic curve comes with a natural involution called the *hyperelliptic involution*, which we denote by ι . If $P = (x, y)$ is a point of $\mathcal{C}$, then $\iota(P)$ is the point $(x, -y - h(x))$, which is also on $\mathcal{C}$. The points which are fixed under the action of ι are called the *ramification points* and they are actually the Weierstraß points of $\mathcal{C}$.

Just like for elliptic curves, we manipulate the curve via an affine equation, but we virtually always work with the projective model. For that we denote by ∞ the point of $\mathcal{C}$ at infinity, which is unique in the imaginary model. We call *affine points* the points of $\mathcal{C}$ that are not ∞ .

VII.1.2. The Jacobian of a Hyperelliptic Curve. Let $\mathcal{C}$ be a hyperelliptic curve. Similarly to elliptic curves, $\mathcal{C}$ comes with a group associated to it. However, the group law is no longer defined over the curve itself but on another object called the *Jacobian* of $\mathcal{C}$ and denoted by $J_{\mathcal{C}}$. First, we need to consider the group of *divisors* on $\mathcal{C}$: these are the finite formal sums of points of $\mathcal{C}$ over $\mathbb{Z}$:

$$\text{Div}(\mathcal{C}) = \left\{ D = \sum_{P_i \in \mathcal{C}} n_i P_i; \quad n_i \in \mathbb{Z}, \quad n_i = 0 \text{ for almost all } i \right\}.$$

The *degree* of a divisor $D = \sum n_i P_i$ is the sum of the coefficients $\deg D = \sum n_i$ and the set of points P_i for which n_i is different from 0 is called the *support* of D . The set of divisors of degree 0 forms a subgroup of $\text{Div}(\mathcal{C})$.

The function field $K(\mathcal{C})$ of $\mathcal{C}$ is the set of rational functions on $\mathcal{C}$. Formally, $K(\mathcal{C})$ is defined as the following field of fractions:

$$K(\mathcal{C}) = \text{Frac} \left(K[X, Y] / (Y^2 + H(X)Y - F(X)) \right).$$

To each non-zero function φ in $K(\mathcal{C})$ we can associate a divisor $\text{div}(\varphi) = \sum \nu_{P_i}(\varphi) P_i$, where $\nu_{P_i}(\varphi)$ is an integer defined as follows: if φ has a zero at P_i , then $\nu_{P_i}(\varphi)$ is the multiplicity of this zero; if φ has a pole at P_i , $\nu_{P_i}(\varphi)$ is the opposite of the order of this pole. Otherwise, $\nu_{P_i}(\varphi) = 0$. A non-zero function has only finitely many zeroes and poles, therefore almost all the $\nu_{P_i}(\varphi)$ are zero and the divisor $\text{div}(\varphi)$ is well defined. Such a divisor is called a *principal* divisor. Furthermore, it can be shown that the degree of a principal divisor is always zero and that the set of the principal divisors is a subgroup of the group of degree 0 divisors. We define the Jacobian $J_{\mathcal{C}}$ of $\mathcal{C}$ to be the quotient group of the degree 0 divisors by the principal divisors. A

consequence of Riemann–Roch theorem is that every element of the Jacobian can be represented by a divisor of the form

$$D = P_1 + P_2 + \cdots + P_r - r\infty,$$

where, for all i , the point P_i is an affine point of $\mathcal{C}$ and the integer r is less than or equal to the genus. If furthermore we ask that P_i be different from $\iota(P_j)$ for all pairs $i \neq j$, then this representation is unique. Such a divisor, with $P_i \neq \iota(P_j)$ for $i \neq j$, is called *reduced*.

Until now, we did not pay attention to the field of definition of the objects we considered. Usually, the good way to say that an object is defined over a field K is to say that it is invariant under the action of any element of the Galois group $\text{Gal}(\overline{K}/K)$. In the case of divisors, this definition is indeed the one we want in the sense that it makes it possible to define the elements of the Jacobian defined over K as a subgroup of the Jacobian defined over $\overline{K}$. However, this implies in particular that a divisor defined over K can include some points P_i in its support which are not defined over K . In that case all the Galois conjugates of P_i appear within D with the same multiplicities as P_i . From the practical point of view, it is annoying to have to work in extensions of the base fields, even when working with elements which are defined over the base field. The fact that a point always comes with its conjugates leads to the following representation of reduced divisors.

PROPOSITION VII.1. *Let $\mathcal{C}$ be a hyperelliptic curve of genus g defined over a field K with equation $Y^2 + H(X)Y = F(X)$. Then the elements of the Jacobian of $\mathcal{C}$ that are defined over K are in one-to-one correspondance with the pairs of polynomials $(a(X), b(X))$ with coefficients in K , such that $\deg(b) < \deg(a) \leq g$, the polynomial a is monic, and a divides $b^2 + bH - F$.*

If a and b are two polynomials that satisfy these conditions, we denote by $\text{div}(a, b)$ the corresponding element of $J_{\mathcal{C}}$.

Let $D = \sum_{1 \leq i \leq r} P_i - r\infty$ be a reduced divisor and denote by (x_i, y_i) the coordinates of P_i . Then the a -polynomial is given by $a(X) = \prod_{1 \leq i \leq r} (X - x_i)$ and the b -polynomial is the only polynomial of degree less than r such that $b(x_i) = y_i$. In the case where the same point occurs several times, the multiplicity must be taken into account to properly define $b(X)$. From now on, when we talk about a reduced divisor, we are actually thinking about this representation with the two polynomials a and b .

The weight of a reduced divisor $D = \text{div}(a, b)$ is the degree of the polynomial a . We denote it by $\text{wt}(D)$, and in the representation

$$D = \sum_{1 \leq i \leq r} P_i - r\infty$$

the weight is given by r . By definition, the weight of a reduced divisor is an integer in $[0, g]$. The neutral element of $J_{\mathcal{C}}$ is the unique weight zero divisor $\text{div}(1, 0)$.

VII.1.3. Hyperelliptic Curves Defined over a Finite Field. When the base field $K = \mathbb{F}_q$ is finite of cardinality q , the set of (a, b) pairs of polynomials that we can form is finite, and therefore the Jacobian is a finite abelian group. Similarly to the case of elliptic curves, the Frobenius endomorphism φ obtained by extending the map $x \mapsto x^q$ to the Jacobian plays a central role in the question of evaluating the cardinalities. Due to the theorem of Weil, the characteristic polynomial of φ is of the form

$$P(T) = T^{2g} + a_1 T^{2g-1} + \cdots + a_g T^g + q a_{g-1} T^{g-1} + \cdots + q^{g-1} a_1 T + q^g,$$

where a_i is an integer for all i . Furthermore, the cardinality of the Jacobian is given by $P(1)$, and all the roots of $P(T)$ have absolute value $\sqrt{q}$. There is also a link between the a_i and the cardinality of $\mathcal{C}$ over the extensions $\mathbb{F}_{q^i}$ for $1 \leq i \leq g$. We skip the details that can be found for instance in [226] or [319] and we give the bounds that we obtain:

THEOREM VII.2. *Let $\mathcal{C}$ be a hyperelliptic curve of genus g defined over a finite field $\mathbb{F}_q$ with q elements. Then we have*

$$(\sqrt{q} - 1)^{2g} \leq \#J_{\mathcal{C}}(\mathbb{F}_q) \leq (\sqrt{q} + 1)^{2g}$$

and

$$|\#\mathcal{C}(\mathbb{F}_q) - (q + 1)| \leq 2g\sqrt{q}.$$

This theorem shows that indeed the Jacobians of hyperelliptic curves defined over a finite field are potentially good candidates for cryptosystems based on the discrete logarithm problem in an abelian group. Indeed, to get a group of size N , it is necessary to have a curve of genus g defined over a finite field with about $N^{1/g}$ elements. If the group law is not too complicated compared to elliptic curves, then the higher number of operations can be somewhat balanced by the fact that they take place in a field whose bit-size is g times smaller.

VII.2. Algorithms for Computing the Group Law

In this section we consider the problem of computing effectively the group law in the Jacobian of a hyperelliptic curve. Having a good algorithm for the group law is crucial when one wants to design a cryptosystem based on hyperelliptic curves.

In all this section, we consider a hyperelliptic curve $\mathcal{C}$ of genus g defined over a field K (not necessarily finite) by the equation $Y^2 + H(X)Y = F(X)$.

VII.2.1. Algorithms for Small Genus. The first algorithm to compute in $J_{\mathcal{C}}$ is due to Cantor [59] and is a translation of Gauss's composition and reduction algorithms for quadratic forms. Here we give another algorithm described in [275] and which goes back to Lagrange. In general this variant saves many operations in the reduction phase. However, for genus 2 this is exactly Cantor's algorithm.

ALGORITHM VII.1: Lagrange's Algorithm for Hyperelliptic Group Law

INPUT: Two reduced divisors $D_1 = \text{div}(a_1, b_1)$ and $D_2 = \text{div}(a_2, b_2)$.

OUTPUT: The reduced divisor $D_3 = \text{div}(a_3, b_3) = D_1 + D_2$.

Composition.

1. Use two extended GCD computations, to get

$$d = \gcd(a_1, a_2, b_1 + b_2 + H) = u_1 a_1 + u_2 a_2 + u_3 (b_1 + b_2 + H).$$
2. $a_3 \leftarrow a_1 a_2 / d^2$
3. $b_3 \leftarrow b_1 + (u_1 a_1 (b_2 - b_1) + u_3 (F - b_1^2 - b_1 H)) / d \pmod{a_3}$
4. If $\deg(a_3) \leq g$ then return $\text{div}(a_3, b_3)$.

Reduction.

5. $\tilde{a}_3 \leftarrow a_3, \tilde{b}_3 \leftarrow b_3$.
 6. $a_3 \leftarrow (F - b_3 H - b_3^2) / a_3$.
 7. $q, b_3 \leftarrow \text{Quotrem}(-b_3 - H, a_3)$.
 8. While $\deg(a_3) > g$ do
 9. $t \leftarrow \tilde{a}_3 + q(b_3 - \tilde{b}_3)$.
 10. $\tilde{b}_3 \leftarrow b_3, \tilde{a}_3 = a_3, a_3 \leftarrow t$.
 11. $q, b_3 \leftarrow \text{Quotrem}(-b_3 - H, a_3)$.
 12. Return $\text{div}(a_3, b_3)$.
-

The important case where $D_1 = D_2$ must be treated separately; indeed some operations can be saved. For instance, we readily know that $\gcd(a_1, a_2) = a_1$.

To analyze the run time of this algorithm, we evaluate how the number of base field operations grows with the genus. The composition steps require one to perform the Euclidean algorithm on polynomials of degree at most g and a constant number of basic operations with polynomials of degree at most $2g$. This can be done at a cost of $O(g^2)$ operations in the base field. The cost of the reduction phase is less obvious. If the base field is large enough compared to the genus, each division is expected to generate a remainder of degree exactly one less than the modulus. The quotients q involved in the algorithm are then always of degree one and one quickly see that the cost is again $O(g^2)$ operations in the base field for the whole reduction process. It can be shown [317] that even if the degrees of the quotients behave in an erratic way, the heavy cost of a step where the quotient is large is amortized by the fact that we have less steps to do and finally the overall cost of reduction is always $O(g^2)$ base field operations.

VII.2.2. Asymptotically Fast Algorithms. In the late eighties, Shanks [298] proposed a variant of Gauss's algorithm for composing positive definite quadratic forms. The main feature of this algorithm, called NUCOMP, is that most of the operations can be done with integers half the size of the integers that are required in the classical algorithm. This makes a great difference if

we are in the range where applying NUCOMP allows us to use machine word sized integers instead of multiprecision integers.

In the classical algorithm, just like in Lagrange's algorithm, the composition step produces large elements that are subsequently reduced. The idea of the NUCOMP algorithm is to more or less merge the composition and the reduction phases, in order to avoid those large elements. A key ingredient is the partial GCD algorithm (also known as the Half GCD algorithm, which is related to the continued fractions algorithm). This is essentially a Euclidean algorithm that we stop in the middle. In NUCOMP we need a partial extended version of it, where only one of the coefficients of the representation is computed.

In [182], it is shown that the NUCOMP algorithm can be adapted to the case of the group law of a hyperelliptic curve. It suffices more or less to replace integers by polynomials. We recall the Partial Extended Half GCD algorithm here for completeness:

ALGORITHM VII.2: Partial Extended Half GCD (PartExtHGCD)

INPUT: Two polynomials A and B of degree $\leq n$ and a bound m

OUTPUT: Four polynomials a, b, α, β such that
 $a \equiv \alpha A \pmod{B}$ and $b \equiv \beta A \pmod{B}$
 and $\deg a$ and $\deg b$ are bounded by m .

An integer z recording the number of operations

1. $\alpha \leftarrow 1, \beta \leftarrow 0, z \leftarrow 0, a \leftarrow A \bmod B, b \leftarrow B.$
 2. While $\deg b > m$ and $a \neq 0$ do
 3. $q, t \leftarrow \text{Quotrem}(b, a).$
 4. $b \leftarrow a, a \leftarrow t, t \leftarrow \beta - q\alpha, \beta \leftarrow \alpha. \quad \alpha \leftarrow t.$
 5. $z \leftarrow z + 1.$
 6. If $a = 0$ then
 7. Return $(A \bmod B, B, 1, 0, 0).$
 8. If z is odd then
 9. $\beta \leftarrow -\beta, b \leftarrow -b.$
 10. Return $(a, b, \alpha, \beta, z).$
-

We are now ready to give the NUCOMP algorithm as described in [182]. We restrict ourselves to the case of odd characteristic where the equation of the curve can be put in the form $Y^2 = F(X)$. The minor modifications to handle the characteristic 2 case can be found in [182].

ALGORITHM VII.3: NUCOMP Algorithm

INPUT: Two reduced divisors $D_1 = \text{div}(u_1, v_1)$ and $D_2 = \text{div}(u_2, v_2)$.

OUTPUT: The reduced divisor $D_3 = D_1 + D_2$.

1. $w_1 \leftarrow (v_1^2 - F)/u_1$, $w_2 \leftarrow (v_2^2 - F)/u_2$.
 2. If $\deg w_2 < \deg w_1$ then exchange D_1 and D_2 .
 3. $s \leftarrow v_1 + v_2$, $m \leftarrow v_2 - v_1$.
 4. By Extended GCD, compute G_0 , b , c such that

$$G_0 = \gcd(u_2, u_1) = bu_2 + cu_1.$$
 5. If $G_0 \nmid s$ then
 6. By Extended GCD, compute G , x , y such that

$$G = \gcd(G_0, s) = xG_0 + ys.$$
 7. $H \leftarrow G_0/G$, $B_y \leftarrow u_1/G$, $C_y \leftarrow u_2/G$, $D_y \leftarrow s/G$.
 8. $l \leftarrow y(bw_1 + cw_2) \bmod H$, $B_x \leftarrow b(m/H) + l(B_y/H)$.
 9. Else
 10. $G \leftarrow G_0$, $A_x \leftarrow G$, $B_x \leftarrow mb$.
 11. $B_y \leftarrow u_1/G$, $C_y \leftarrow u_2/G$, $D_y \leftarrow s/G$.
 12. $(b_x, b_y, x, y, z) \leftarrow \text{PARTEXTHGCD}(B_x \bmod B_y, B_y, (g+2)/2)$.
 13. $a_x \leftarrow Gx$, $a_y \leftarrow Gy$.
 14. If $z = 0$ then
 15. $u_3 \leftarrow b_y C_y$, $Q_1 \leftarrow C_y b_x$, $v_3 \leftarrow v_2 - Q_1$.
 16. Else
 17. $c_x \leftarrow (C_y b_x - mx)/B_y$, $Q_1 \leftarrow b_y c_x$, $Q_2 \leftarrow Q_1 + m$.
 18. $d_x \leftarrow (D_y b_x - w_2 x)/B_y$, $Q_3 \leftarrow y d_x$, $Q_4 \leftarrow Q_3 + D_y$.
 19. $d_y \leftarrow Q_4/x$, $c_y \leftarrow Q_2/b_x$.
 20. $u_3 \leftarrow b_y c_y - a_y d_y$, $v_3 \leftarrow (G(Q_3 + Q_4) - Q_1 - Q_2)/2 \bmod u_3$.
 21. While $\deg(u_3) > g$ do
 22. $u_3 \leftarrow (v_3^2 - F)/u_3$, $v_3 \leftarrow -v_3 \bmod u_3$.
 23. Return $\text{div}(u_3, v_3)$.
-

In the NUCOMP algorithm, if the input divisors are distinct, then it is likely that $G_0 = 1$, and then Steps 5–8 are skipped. On the other hand, in the case of doubling a divisor, these steps are always performed whereas Step 4 is trivial. Hence it is interesting to have another version of NUCOMP that is specific to doubling. Shanks had such an algorithm which he called NUDPL and it has been extended to our case in [182].

Steps 21–22 are there to adjust the result in the case where the divisor $\text{div}(u_3, v_3)$ is not completely reduced at the end of the main part of the algorithm. In general, at most one or two loops are required. Therefore the NUCOMP algorithm requires a constant number of Euclidean algorithms on polynomials of degree $O(g)$ and a constant number of basic operations with polynomials of degree $O(g)$. Thus the complexity is $O(g^2)$, again, just as for

Lagrange's algorithm. However, as we said before, the NUCOMP algorithm works with polynomials of smaller degree (in practice rarely larger than g).

Up to now we have considered naive algorithms for polynomial multiplications and GCDs. If we used an FFT-based algorithm, two polynomials of degree $O(n)$ can be multiplied in time $O(n \log n \log \log n)$ operations in the base field. It is then possible to design a recursive GCD algorithm that requires $O(n \log^2 n \log \log n)$ operations [141]. Furthermore, the same algorithm can also be used to compute the Partial Extended Half GCD in the same run time (this is actually a building block for the fast recursive GCD computation). Plugging those algorithms in NUCOMP and NUDUPL yields an overall complexity of $O(g \log^2 g \log \log g)$ which is asymptotically faster than the $O(g^2)$ complexity of Lagrange's algorithm.

VII.2.3. Which Algorithm in Practice. When the genus is large, the best algorithm is not the same as when the genus is small. In particular, for very small genus, it is possible to write down the explicit formulae corresponding to the Cantor or Lagrange algorithm. Then some operations can be saved by computing only the coefficients that are necessary. For instance, the quotient of two polynomials costs less if it is known in advance that the division is exact (see [258]). For genus 2 and 3, such optimized formulae have been worked out by many contributors. We refer to [212] for a survey of the current situation in genus 2, and to [277] for genus 3. In the case of genus 2, the best formulae for adding (resp. doubling) have a cost of 25 multiplications and 1 inversion (resp. 27 multiplications and 1 inversion).

When the genus is larger than 3, explicit formulae are not available and the best algorithm is then Lagrange's algorithm.

According to [182], the genus for which the NUCOMP and NUDUPL algorithms are faster than Lagrange's algorithm is around 10. This value depends highly on the implementation of those two algorithms, on the particular type of processor that is used, and on the size of the base field.

It is likely that the asymptotically fast algorithm for computing GCD will never be useful for practical applications: the point at which it becomes faster than the classical Euclidean algorithm is around genus 1000.

We finally mention that an analogue of projective coordinates for elliptic curves has been designed. This allows one to perform the group law without any inversion, which can be crucial in some constrained environments such as smart cards. In addition, in the spirit of [85], different types of coordinates can be mixed to have a faster exponentiation algorithm [212].

VII.3. Classical Algorithms for HCDLP

The hyperelliptic curve discrete logarithm problem (HCDLP) is a generalization of the elliptic curve discrete logarithm problem (ECDLP). As such it comes as no surprise that every known attack on the ECDLP, for some

particular instances, can be extended to an attack on the HCDLP, for the corresponding weak instances. We review all these attacks that we call classical, in opposition to the index-calculus attacks which are specific to hyperelliptic curves. We assume that the reader is familiar with all attacks on ECDLP that are listed in Chapter V of [ECC].

VII.3.1. The Simplification of Pohlig and Hellman. Let G denote an abelian group of known order N whose factorization is also known. The algorithm of Pohlig and Hellman reduces the computation of a discrete logarithm in G to $O(\log N)$ computations of discrete logarithms in subgroups of G of order p where p is a prime factor of N . The method works for a generic group and in particular can be applied to HCDLP. It means that solving a discrete logarithm problem in the Jacobian of a hyperelliptic curve is not harder than the discrete logarithm problem in the largest prime order subgroup of the Jacobian.

VII.3.2. The MOV and Frey–Rück Attacks. Let $\mathcal{C}$ be a hyperelliptic curve of genus g over the finite field $\mathbb{F}_q$ and let us consider a DL problem in a subgroup of $J_{\mathcal{C}}$ of prime order m . Denote by k the smallest integer such that m divides $q^k - 1$, i.e., k is the order of q modulo m . Then there is an efficient algorithm that converts the initial discrete logarithm problem into a discrete logarithm problem in the multiplicative group of the finite field $\mathbb{F}_{q^k}$. If k is small enough, the existence of a subexponential algorithm for computing discrete logarithms in finite fields can lead to an efficient way of solving the initial problem in $J_{\mathcal{C}}$.

For more details on the method to transport the HCDLP to a classical DLP we refer to Chapter IX. Indeed the key ingredients are the Weil and Tate pairings which can, as we shall see in Chapter X, also be used in a constructive way.

VII.3.3. The Rück Attack. The previous attack works only if the order m of the subgroup is prime to the field characteristic (otherwise the parameter k is not even defined). In the case where m is precisely the field characteristic, Rück [283] has extended the so-called anomalous curves attack (see [ECC, Section V.3]) to hyperelliptic curves. He proved

THEOREM VII.3. *Let $\mathcal{C}$ be a hyperelliptic curve of genus g defined over a finite field of characteristic p . The discrete logarithm problem in a subgroup of $J_{\mathcal{C}}$ of order p can be solved with at most $O(\log p)$ operations in the base field and in the Jacobian.*

In fact Rück proved his result in a more general setting where the curve does not need to be hyperelliptic.

The principle is the following: let D be a degree 0 divisor whose class in $J_{\mathcal{C}}$ is of order p . Then, by definition, there exists a function φ such that $\text{div}(\varphi) = p \cdot D$. The holomorphic differential $d\varphi/\varphi$ can be expressed using a

local parameter t at the point at infinity ∞ : we can write $d\varphi/\varphi = \frac{\partial\varphi/\partial t}{\varphi}dt$. We can then define the power series

$$\frac{\partial\varphi/\partial t}{\varphi} = \sum_{i=0}^{\infty} a_i t^i,$$

where the a_i are scalars. The key point is that the function that maps the class of D to the $2g - 1$ first coefficients $(a_0, \dots, a_{2g-2})$ is an homomorphism of additive groups. The discrete logarithm problem is then transported into a group where it is easily solvable.

The explicit computation of this homomorphism can be done efficiently using a technique similar to the method used for computing the Weil pairing.

VII.3.4. Baby Step/Giant Step and Random Walks. The generic algorithms, such as the Baby Step/Giant Step algorithm, as described in Section V.4 of [ECC], can be applied to compute discrete logarithms in any abelian group. In particular, it is straightforward to obtain an algorithm for solving the HCDLP in a subgroup of order N of the Jacobian of a hyperelliptic curve in at most $O(\sqrt{N})$ group operations.

Also, the parallelized Pollard lambda method of van Oorschot and Wiener [266] directly applies to HCDLP, providing a $O(\sqrt{N})$ algorithm with a speed-up by a factor of M if M machines are used and requiring a small amount of memory.

Furthermore the presence of equivalence classes arising from the hyperelliptic involution as well as any other quickly computable automorphism can be used just like for elliptic curves [111], [138].

VII.4. Smooth Divisors

In 1994, Adleman, DeMarrais and Huang [4] discovered a subexponential algorithm for computing discrete logarithms in the Jacobian of hyperelliptic curves of “large genus.” Compared with points of elliptic curves, divisors in the Jacobian of a hyperelliptic curve carry more structure. Indeed, a reduced divisor is represented using two polynomials (a, b) , and the factorization of a as a polynomial in $K[X]$ is somewhat compatible with the Jacobian group law. This fact is the key stone for defining a smoothness notion and then the Adleman, DeMarrais and Huang index-calculus algorithm.

LEMMA VII.4. *Let $\mathcal{C}$ be a hyperelliptic curve over a finite field $\mathbb{F}_q$. Let $D = \text{div}(a, b)$ be a reduced divisor in $J_{\mathcal{C}}(\mathbb{F}_q)$. Write $a(X) = \prod a_i(X)$ as the factorization in irreducible factors of $a(X)$ in $\mathbb{F}_q[X]$. Let $b_i(X) = b(X) \pmod{a_i(X)}$. Then $D_i = \text{div}(a_i, b_i)$ is a reduced divisor and $D = \sum D_i$ in $J_{\mathcal{C}}$.*

With this result, a reduced divisor can be rewritten as the sum of reduced divisors of smaller size, in the sense that the degree of their a -polynomial is

smaller. More precisely, we have $\text{wt}(D) = \sum \text{wt}(D_i)$. The divisors that can not be rewritten are those for which the a -polynomial is irreducible.

DEFINITION VII.5. *A reduced divisor $D = \text{div}(a, b)$ is said to be prime if the polynomial a is irreducible.*

DEFINITION VII.6. *Let B be an integer. A reduced divisor $D = \text{div}(a, b)$ is said to be B -smooth if the polynomial a is the product of polynomials of degree at most B .*

This definition is very analogous to the smoothness notion that is used to design index-calculus algorithms in the multiplicative group of a finite field. The next step is to study the proportion of smooth divisors.

We start with the easy case where $B = 1$, which will be of great importance in applications.

PROPOSITION VII.7. *Let $\mathcal{C}$ be a hyperelliptic curve of genus g over the finite field $\mathbb{F}_q$. Then the proportion of 1-smooth reduced divisors in $J_{\mathcal{C}}(\mathbb{F}_q)$ tends to $1/g!$ when g is fixed and q tends to infinity.*

The proof relies on Theorem VII.2. The 1-smooth reduced divisors are formed using at most g points of $\mathcal{C}(\mathbb{F}_q)$, and the order does not matter. Therefore there are about $q^g/g!$ reduced divisors that are 1-smooth. This number should be compared with $\#J_{\mathcal{C}}(\mathbb{F}_q) \approx q^g$. The error terms given by the precise bounds are an order of magnitude smaller, so that we obtain the announced proportion.

If we do not fix $B = 1$, intuitively we expect the same kind of behaviour as for polynomials over finite fields for which taking a subexponential smoothness bound yields a subexponential proportion of smooth elements. This heuristic has been made precise by Enge and Stein [118]. We recall the definition of the so-called subexponential function parametrized by a real α in $[0, 1]$ and a constant c :

$$L_N(\alpha, c) = \exp \left(c(\log N)^\alpha (\log \log N)^{1-\alpha} \right).$$

THEOREM VII.8 (Engel and Stein). *Let $\mathcal{C}$ be a hyperelliptic curve of genus g defined over the finite field $\mathbb{F}_q$. Let ρ be a positive constant and let $B = \lceil \log_q L_{q^g}(\frac{1}{2}, \rho) \rceil$ be a smoothness bound. Then the number of B -smooth reduced divisors in $J_{\mathcal{C}}(\mathbb{F}_q)$ is at least*

$$q^g \cdot L_{q^g} \left(\frac{1}{2}, -\frac{1}{2\rho} + o(1) \right).$$

It is worth noting that B is defined as the *ceiling* of the usual expression. Indeed, as a degree, it has to be an integer. This looks inoffensive but is actually the heart of the main problem that occurs below, namely the subexponential behaviour can be observed only if the genus is large enough compared to the size of the base field. If the genus is too small, the smoothness bound would be smaller than 1, and we are back to the situation of the previous proposition.

VII.5. Index-Calculus Algorithm for Hyperelliptic Curves

The original algorithm by Adleman, DeMarrais and Huang is highly heuristic and more theoretical than of practical use. Their idea was to work with functions on the curve. This idea was pushed further by Flassenberg and Paulus [121] who were able to design a sieving technique, and they did the first implementation of this kind of algorithm. Later on, Müller, Stein and Thiel [257] and Enge [116] managed to get rid of the heuristics, by working with elements of the Jacobian instead of functions, so that Theorem VII.8 could be applied.

The method we shall now describe is the further improvement by Enge and Gaudry [117]. We give a version which is close to what is really done when it is implemented; there exists also a more complicated version that has a heuristic-free complexity.

We first present a sketch of the algorithm, and then details about each phase are given.

ALGORITHM VII.4: Hyperelliptic Index-Calculus Algorithm

INPUT: A divisor D_1 in $J_C(\mathbb{F}_q)$ with known order $N = \text{ord}(D_1)$,
and a divisor $D_2 \in \langle D_1 \rangle$

OUTPUT: An integer λ such that $D_2 = \lambda D_1$

1. Fix a smoothness bound and construct the factor basis.
 2. While not enough relations have been found do:
 3. Pick a random element $R = \alpha D_1 + \beta D_2$.
 4. If R is smooth, record the corresponding relation.
 5. Solve the linear algebra system over $\mathbb{Z}/N\mathbb{Z}$.
 6. Return λ .
-

VII.5.1. Construction of the Factor Basis. First a smoothness bound B is chosen. In the complexity analysis, we shall explain how to get the best value for B . Then the factor basis $\mathcal{F}$ contains all the prime reduced divisors of weight at most B :

$$\mathcal{F} = \{P \in J_C : P \text{ is prime, } \text{wt}(P) \leq B\}.$$

This set can be constructed in a naive way: for each monic polynomial of degree at most B , check if it is irreducible and if it is the a -polynomial of a reduced divisor. In that case, find all the compatible b -polynomials and add the corresponding divisors to $\mathcal{F}$. For convenience, we give names to the elements of $\mathcal{F}$:

$$\mathcal{F} = \{g_i : i \in [1, \#\mathcal{F}]\}.$$

VII.5.2. A Pseudo-Random Walk. Selecting a random element $R = \alpha D_1 + \beta D_2$ is costly: the values of α and β are randomly chosen in the interval $[1, N]$ and then two scalar multiplications have to be done. Using the binary powering method, the cost is $O(\log N)$ group operations. We use a pseudo-random walk instead, so that the construction of each new random element costs just one group operation.

The pseudo-random walk is exactly the same as the one which is used in [323] in discrete log algorithms based on the birthday paradox. For j from 1 to 20 we randomly choose a_j and b_j in $[1, N]$ and compute the “multiplier” $T_j \leftarrow a_j D_1 + b_j D_2$. In our case where the group is traditionally written additively, “summand” would be a better name but we stick to the classical name. The pseudo-random walk is then defined as follows:

- R_0 is given by $\alpha_0 D_1 + \beta_0 D_2$ where α_0 and β_0 are randomly chosen in $[1, N]$.
- R_{i+1} is defined by adding to R_i one of the multipliers T_j . The index j is given by the value of a hash function evaluated at R_i . Then the representation of R_{i+1} in terms of D_1 and D_2 is deduced from the corresponding representation of R_i : we set $\alpha_{i+1} = \alpha_i + a_j$ and $\beta_{i+1} = \beta_i + b_j$ for the same j .

Practical experiments [323] suggest that by taking 20 multipliers the pseudo-random walk behaves almost like a purely random walk. In our case, it is not really necessary to take j deterministic in terms of R_i : picking a random multiplier at each step would do the job. The determinism in the pseudo-random walk was necessary in [323] to use distinguished points, but it is not used in our algorithm.

VII.5.3. Collection of Relations. Each time we produce a new random element $R = \alpha D_1 + \beta D_2$, we test it for smoothness. If R is not smooth, then we continue the pseudo-random walk to get another element. The smoothness test is done by factoring the a -polynomial of R . If all its irreducible factors have degree at most B , then Lemma VII.4 allows us to write R as a sum of elements of $\mathcal{F}$. The k th smooth element $S_k = \alpha_k D_1 + \beta_k D_2$ that is found is stored in the k th column of a matrix $M = (m_{ik})$ which has $\#\mathcal{F}$ rows:

$$S_k = \sum_{1 \leq i \leq \#\mathcal{F}} m_{ik} g_i = \alpha_k D_1 + \beta_k D_2.$$

We also store the values of α_k and β_k for later use.

VII.5.4. Linear Algebra. After $\#\mathcal{F} + 1$ relations have been collected, the matrix M has more columns than rows. As a consequence, there exists a non-trivial vector (γ_k) in its kernel. Then by construction we have

$$\sum \gamma_k S_k = 0 = \left(\sum \gamma_k \alpha_k \right) D_1 + \left(\sum \gamma_k \beta_k \right) D_2.$$

Now, with high probability, $\sum \gamma_k \beta_k$ is different from 0 modulo N and we can compute the wanted discrete logarithm

$$\lambda = -(\sum \alpha_k \gamma_k) / (\sum \beta_k \gamma_k) \pmod{N}.$$

To compute the vector (γ_k) , we could use Gaussian elimination. However, we can do better, because the matrix is very sparse. Indeed, in the k th column, a non-zero entry at index i means that the i th element of $\mathcal{F}$ has a a -polynomial which divides the a -polynomial of the reduced divisor S_k , the degree of which is bounded by g , the genus of $\mathcal{C}$. Hence there are at most g non-zero entries in each column. For such sparse matrices, Lanczos's algorithm or Wiedemann's algorithm are much faster than Gaussian elimination, though they are not deterministic.

VII.6. Complexity Analysis

In our analysis, we denote by c_J the cost of one group operation in $J_{\mathcal{C}}(\mathbb{F}_q)$, by c_F the cost of testing the B -smoothness of a polynomial of degree g over $\mathbb{F}_q$ and by c_N the cost of an operation in $\mathbb{Z}/N\mathbb{Z}$. With Lagrange's algorithm and naive multiplication algorithms in $\mathbb{F}_q$, we obtain $c_J = O(g^2 \log^2 q)$, and if we use NUCOMP and asymptotically fast algorithms everywhere, we obtain $c_J = O(g \log q \varepsilon(g, \log q))$, where $\varepsilon(g, \log q)$ is bounded by a polynomial in $\log g$ and $\log \log q$. The B -smoothness test is the first phase of the Cantor-Zassenhaus algorithm called Distinct-Degree-Factorization, and its costs is $c_F = O(Bg^2 \log^3 q)$ if naive multiplication algorithms are used.

VII.6.1. Analysis When g is Large Enough. We write the smoothness bound B in the form

$$B = \left\lceil \log_q L_{q^g} \left(\frac{1}{2}, \rho \right) \right\rceil,$$

where ρ is some positive real number. By Theorem VII.8, the number of B -smooth reduced divisors in $J_{\mathcal{C}}(\mathbb{F}_q)$ is at least

$$q^g \cdot L_{q^g} \left(\frac{1}{2}, -\frac{1}{2\rho} \right),$$

so that the expected number of steps in the pseudo-random walk before finding a B -smooth divisor is about $L_{q^g} \left(\frac{1}{2}, \frac{1}{2\rho} \right)$. We need $\#\mathcal{F} + 1$ relations, therefore the expected number of steps in the pseudo-random walk that are required to fill in the matrix is about

$$\#\mathcal{F} \cdot L_{q^g} \left(\frac{1}{2}, \frac{1}{2\rho} \right).$$

The cardinality of the factor basis is bounded by a constant times the number of polynomials of degree at most B . Hence $\#\mathcal{F}$ is $O(q^B) = O(L_{q^g}(\frac{1}{2}, \rho))$.

Putting all of this together, we see that the algorithm requires

$$O\left(L_{q^g}\left(\frac{1}{2}, \rho + \frac{1}{2\rho}\right)\right)$$

steps in the pseudo-random walk. At each step, we need to perform one group operation (which costs c_J) and the factorization of the a -polynomial of the divisor (which costs c_F).

Once the matrix M has been computed, we need to find a vector of the kernel of M . This matrix has size $\#\mathcal{F}$ and has at most g non-zero entries per column. The cost of Lanczos's algorithm is then

$$O((\#\mathcal{F})^2 g) = O\left(g L_{q^g}\left(\frac{1}{2}, 2\rho\right)\right)$$

operations in $\mathbb{Z}/N\mathbb{Z}$.

Thus the total run time of the algorithm is

$$O\left((c_J + c_F) \cdot L_{q^g}\left(\frac{1}{2}, \rho + \frac{1}{2\rho}\right) + (gc_N) \cdot L_{q^g}\left(\frac{1}{2}, 2\rho\right)\right).$$

To analyze this run time and find the optimal value for ρ , we remark that c_J , c_F , c_N and g are all polynomials in g and $\log q$, whereas the $L_{q^g}(\dots)$ parts are subexponential in those parameters. Therefore it makes sense to neglect them when trying to balance both phases. We are then left with minimizing the function

$$L_{q^g}\left(\frac{1}{2}, \rho + \frac{1}{2\rho}\right) + L_{q^g}\left(\frac{1}{2}, 2\rho\right) \approx L_{q^g}\left(\frac{1}{2}, \max(\rho + \frac{1}{2\rho}, 2\rho)\right).$$

An easy calculation shows that the minimum is obtained for $\rho = \frac{1}{\sqrt{2}}$. With this choice of ρ , we obtain an overall complexity of

$$O\left((c_J + c_F + gc_N) \cdot L_{q^g}\left(\frac{1}{2}, \sqrt{2}\right)\right).$$

Before putting this result into a theorem, we need to make clear an abusive simplification that we did along the way: when evaluating the size of the factor basis, we “forgot” the ceiling sign for B . However, if $\log_q L_{q^g}(\frac{1}{2}, \sqrt{2})$ is close to 0, the fact that we round it to 1 introduces an exponential factor in the complexity analysis. Therefore, what we did makes sense only if this value is large enough so that the rounding has no effect on the complexity. This requirement is fulfilled if and only if g is large enough compared to $\log q$.

THEOREM VII.9 (Enge–Gaudry[117]). *There exists an algorithm that can solve the HCDLP in time $O\left(L_{q^g}(\frac{1}{2}, \sqrt{2})\right)$ when q and g grow to infinity with the condition that g stays “large enough” compared to $\log q$.*

The complexity of the algorithm can be made more precise by assuming that there exists a constant ϑ such that $g > \vartheta \log q$. Then the run time is in

$O(L_{q^g}(\frac{1}{2}, c(\vartheta)))$, where $c(\vartheta)$ is an explicit function that tends to $\sqrt{2}$ when ϑ tends to infinity.

VII.6.2. Complexity When g is Fixed. In this subsection, we analyze the behaviour of Algorithm VII.4 when g is not large compared to $\log q$, and in particular when g is fixed and q tends to infinity. In that case, the theoretical optimal smoothness would be $\log_q L_{q^g}(\frac{1}{2}, \sqrt{2})$, which tends to 0. Therefore we assume from now on that $B = 1$.

By Proposition VII.7, the probability for a divisor to be 1-smooth is $\frac{1}{g!}$. Note that in our algorithm, we take random elements not in the whole Jacobian but in the subgroup generated by D_1 . Heuristically, as long as the subgroup is large enough compared to $g!$, the smoothness probability of $\frac{1}{g!}$ still holds.

The expected time for finding enough relations to fill in the matrix is then $(c_J + c_F)g!q$, and the time for the linear algebra is $O(c_N g q^2)$. Putting this together, we obtain

THEOREM VII.10 (Gaudry[142]). *There exists an algorithm that can solve the HCDLP in large enough subgroups in time $O((g^2 \log^3 q)g!q + (g^2 \log q)q^2)$, when q tends to infinity and g is fixed.*

VII.6.3. Very Small Genera. In the case where the genus is fixed and very small, Harley proposed an improvement to the previous technique. Taking $B = 1$ is far from the theoretical optimal value for subexponentiality, but as a degree it cannot be reduced. Still it is possible to keep in the basis only a small proportion of the weight one reduced divisors. Fix $\#\mathcal{F} = \frac{q}{\alpha}$, where $\alpha < q$ is to be determined. A 1-smooth element has a probability $1/\alpha^g$ to be represented over this reduced basis, therefore the expected number of steps before finding a relation is $\alpha^g g!$. On the other hand, the number of required relations is also reduced: we need only $q/\alpha + 1$ columns in the matrix. Therefore the expected number of steps in order to fill in the matrix is $qg!\alpha^{g-1}$. Also, the linear algebra step is made easier by the fact that the factor basis is smaller, the cost is $O(q^2/\alpha^2)$. Balancing the two phases leads to an optimal value for α which is $(q/g!)^{1/(g+1)}$. With this choice, we obtain an overall complexity of (forgetting the coefficients which are polynomials in g and $\log q$)

$$O\left(q^{\frac{2g}{g+1}} \cdot (g!)^{\frac{2}{g+1}}\right).$$

The term in $g!$ has a negligible contribution, especially because g is supposed to be small. What remains is then a complexity of $O(q^{\frac{2g}{g+1}})$.

This idea of reducing the factor basis has been pushed further by Thériault [325]: by considering some of the elements as “large primes,” the complexity can be reduced to $O(q^{2-\frac{4}{2g+1}})$. The practical use of this last version is still to be studied.

In the following table, we compare the complexities of these two variants to the complexity of Pollard rho algorithm, which has a complexity in the square root of the group size, namely $O(q^{g/2})$.

g	1	2	3	4	5	6	7	8
rho	$q^{1/2}$	q	$q^{3/2}$	q^2	$q^{5/2}$	q^3	$q^{7/2}$	q^4
Index	q	$q^{4/3}$	$q^{3/2}$	$q^{8/5}$	$q^{5/3}$	$q^{12/7}$	$q^{7/4}$	$q^{16/9}$
Thériault	$q^{2/3}$	$q^{6/5}$	$q^{10/7}$	$q^{14/9}$	$q^{18/11}$	$q^{22/13}$	$q^{26/15}$	$q^{30/17}$

We conclude that for genus 4 curves, the index-calculus method provides a faster attack than the generic Pollard rho algorithm, and using Thériault's method, even genus 3 curves seem to be subject to the attack.

VII.7. Practical Considerations

We mention a few improvements that do not change the complexity but that can make big differences in a practical implementation. First of all it is important that the basic arithmetic be as fast as possible. Once the Jacobian group law, the polynomial factorization and the arithmetic in $\mathbb{Z}/N\mathbb{Z}$ have been optimized, here are some more tricks.

- A hyperelliptic curve always comes with its hyperelliptic involution which can be computed almost for free. As a consequence, there is no need to store both a divisor and its opposite in the factor basis: if the opposite occurs in the smooth divisor, just put a -1 instead of a 1 in the matrix. More generally, if the curve has other cheap automorphisms (for instance if this is a Koblitze curve), then only one element in each orbit needs to be stored in $\mathcal{F}$.
- The search for relations is trivially parallelized by taking a different random walk on each processor.
- As said above, testing if a divisor is smooth does not require the full factorization of the a -polynomial. In fact, in principle, a squarefreeness test has to be run before going to the Distinct-Degree-Factorization step. But in our case, this additional test is not necessary: non-squarefree polynomials are rare and saving this computation more than compensates the loss of these potentially smooth divisors.
- In the case of the reduced basis (when the genus is very small), we can choose to keep in the basis the elements with small abscissa. Then a reduced divisor of degree g can be written on this basis only if its degree $g - 1$ coefficient is small. Indeed, the divisor is smooth if and only if its a -polynomial splits completely and all its roots are small, and the degree $g - 1$ coefficient is the sum of the roots. This criterion is much faster than the Distinct Degree Factorization algorithm and enables one to eliminate most of the non-smooth divisors.
- If the group law is more costly than the smoothness test, it can be sped-up as follows: between each step of the classical random walk, we

introduce small steps where we add an element of the factor basis to the current element. We continue the small steps until a smooth divisor is found. Then we can write a relation in the matrix. The drawback is that there is one more term in the matrix corresponding to the multiple of the element that we added. On the other hand, most of the time is spent in small steps in which we add a weight one divisor. Therefore the group law is much faster.

- The implementation of a fast sparse linear algebra algorithm is not a simple task. Our case is different from the integer factorization case, where the matrix is defined over $\mathbb{F}_2$. Here, the base ring is not that small and most of the tricks do not apply. If parallelization is not necessary, then Lanczos's algorithm is probably the best choice. Otherwise the Block-Wiedemann algorithm can give better results (see [328] for recent experiments).
- If the genus is large, so that the optimal B is not one, a precise non-asymptotical analysis must be done to find the best value for B . This was done in [181].

CHAPTER VIII

Weil Descent Attacks

F. Hess

Weil descent attacks provide means of attacking the DLP for elliptic curves, or for more general algebraic curves such as hyperelliptic curves, when they are used over finite extension fields, i.e., non-prime fields.

The application of the original basic idea, the Weil restriction of scalars for elliptic curves, to cryptography and the ECDLP was first suggested in [124]. An important step forward was made in [145] using explicit covering techniques, relating the ECDLP to a potentially easier HCDLP. Since then, variations, generalizations and ramifications of the employed methodology have been investigated in some detail.

The aim of this chapter is to explain the basic ideas, to summarize the main results about Weil descent attacks for elliptic curves and to discuss the relevance to ECC.

VIII.1. Introduction — the Weil Descent Methodology

Throughout this chapter we let K/k denote an extension of finite fields of degree n . The characteristic and cardinality of k are p and $q = p^r$, respectively.

VIII.1.1. Curves in the Weil Restriction. Let $\mathcal{E}$ be an elliptic curve over K . The initial motivation for the Weil descent attacks came from the consideration of the Weil restriction $\text{Res}_{K/k}(\mathcal{E})$ of $\mathcal{E}$ with respect to K/k , suggested by Frey [124].

The Weil restriction $\text{Res}_{K/k}(\mathcal{E})$ is an abelian variety of dimension n defined over k , as opposed to $\mathcal{E}$, which is an abelian variety of dimension one over K . The group $\text{Res}_{K/k}(\mathcal{E})(k)$ of k -rational points of $\text{Res}_{K/k}(\mathcal{E})$ is isomorphic to the group $\mathcal{E}(K)$ of K -rational points of $\mathcal{E}$ and thus contains an equivalent version of any DLP in $\mathcal{E}(K)$. Given $\mathcal{E}$ and K/k and the defining equations, the group law of $\text{Res}_{K/k}(\mathcal{E})$ and the isomorphism of the point groups can be computed without much difficulty. We do not need the details here and refer to [124], [137], [145] instead.

The main idea now is the following. An algebraic curve $\mathcal{C}^0$ and a map $\mathcal{C}^0 \rightarrow \text{Res}_{K/k}(\mathcal{E})$ defined over k lead to a map $\phi : \text{Jac}(\mathcal{C}^0) \rightarrow \text{Res}_{K/k}(\mathcal{E})$, due to the functorial property of $\text{Jac}(\mathcal{C}^0)$. If we take such a curve $\mathcal{C}^0$, we may be able to lift a given DLP from $\text{Res}_{K/k}(\mathcal{E})(k)$ to $\text{Jac}(\mathcal{C}^0)(k)$. The DLP in

$\text{Jac}(\mathcal{C}^0)(k)$ can then be attacked, possibly more efficiently, by index-calculus methods on $\mathcal{C}^0(k)$ than by the Pollard methods on $\mathcal{E}(K)$, the main point being that $\mathcal{C}^0$ is defined over the small field k .

In order to find $\mathcal{C}^0$ one can intersect $\text{Res}_{K/k}(\mathcal{E})$ with suitable hyperplanes, and we remark that there is a fairly natural choice of such a hyperplane. It is then a priori not clear whether the DLP can be lifted to $\text{Jac}(\mathcal{C}^0)(k)$; however some evidence is given by the fact that $\text{Res}_{K/k}(\mathcal{E})$ is simple in many interesting cases [105]. Another quite difficult problem is how to actually lift the DLP to $\text{Jac}(\mathcal{C}^0)(k)$ using explicit equations. We refer to [137] for a more detailed discussion.

VIII.1.2. Covering Techniques. Covering techniques boil down to a reformulation of the method of the previous section at the level of curves, their function fields and Galois theory. Curves are basically one-dimensional objects as opposed to the above Weil restriction and Jacobians, and their function fields can furthermore be viewed as “equationless” substitutes. This leads to a much easier and algorithmically accessible treatment of the previous section and was first applied by Gaudry, Hess and Smart in [145], an approach which is now referred to as the GHS attack.

We consider the following general situation. Let E denote an elliptic function field over K and C a finite extension field of E such that there is a field automorphism σ which extends the Frobenius automorphism of K/k and has order n on C . We say that σ is a Frobenius automorphism of C with respect to K/k and denote the fixed field of σ by C^0 . The extension C/C^0 has degree n and the exact constant field of C^0 is k . Choosing suitable defining equations we can regard E , C and C^0 as the function fields of curves $\mathcal{E}$, $\mathcal{C}$ and $\mathcal{C}^0$, respectively, where $\mathcal{E}$ is an elliptic curve, $\mathcal{E}$ and $\mathcal{C}$ are defined over K and $\mathcal{C}^0$ is defined over k . We denote the divisor class groups of E , C and C^0 by $\text{Pic}_K^0(E)$, $\text{Pic}_K^0(C)$ and $\text{Pic}_k^0(C^0)$ so that $\mathcal{E}(K) \cong \text{Pic}_K^0(E)$ and $\text{Jac}(\mathcal{C}^0)(k) \cong \text{Pic}_k^0(C^0)$. The conorm and norm maps of function field extensions yield homomorphisms $\text{Con}_{C/E} : \text{Pic}_K^0(E) \rightarrow \text{Pic}_K^0(C)$ and $N_{C/C^0} : \text{Pic}_K^0(C) \rightarrow \text{Pic}_k^0(C^0)$. Figure VIII.1 contains a graphical presentation of the situation.

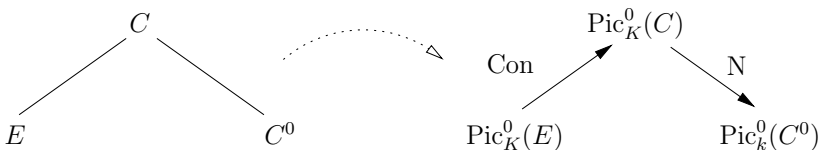


FIGURE VIII.1. Diagram of function fields and divisor class groups

The composition of N_{C/C^0} , $\text{Con}_{C/E}$ and the isomorphism between $\text{Pic}_K^0(E)$ and $\mathcal{E}(K)$ is a homomorphism from $\mathcal{E}(K)$ to $\text{Pic}_k^0(C^0)$ which we denote by ϕ .

Using ϕ , a discrete logarithm problem $Q = \lambda P$ in $\mathcal{E}(K)$ is mapped to the discrete logarithm problem $\phi(Q) = \lambda\phi(P)$ in $\text{Pic}_k^0(C^0)$, where it can be attacked possibly more efficiently than in $\mathcal{E}(K)$, since it is defined over the small field k and index-calculus methods can be applied. Of course, P and Q should not be in the kernel of ϕ . The index-calculus methods depend exponentially or subexponentially on the genus g of C^0 ; see Chapter VII, [92], [117] and [169]. There are thus three main questions.

1. How can such C and σ be constructed?
2. Does ϕ preserve the DLP?
3. Is the genus of C^0 small enough?

The construction of such a C and σ can be achieved quite generally using techniques from Galois theory. For example, one determines a suitable rational subfield $K(x)$ of E and defines C to be the splitting field of E over $k(x)$. A sufficient condition for the existence of a suitable σ is then that n is coprime to the index $[C : K(x)]$. For the sake of efficiency and explicitness, Artin–Schreier and Kummer extensions are the most prominent constructions used.

The question of whether ϕ preserves the DLP can be answered affirmatively in most cases of interest [104], [168].

Finally, the genus can be explicitly determined or estimated for the employed Artin–Schreier and Kummer extensions, given E and n . As it turns out, the genus is in general exponential in n , and it is smaller only in exceptional cases. This is the reason why the Weil descent methodology does not apply to general or randomly chosen elliptic curves when n is large.

These issues are discussed in detail in the following sections.

VIII.1.3. Remarks. The approaches of Subsections VIII.1.1 and VIII.1.2 are jointly carried out in [145]. For a further discussion of the equivalence of these two approaches see [103, Chapter 3] and the appendix of [104].

The term “Weil descent” is actually used with a different meaning in mathematics than we do here and in cryptography. There a “Weil descent argument” refers to a special proof technique about the field of definition of a variety, introduced by A. Weil in [347], while here we loosely mean the transition of an elliptic curve to its Weil restriction and further considerations by which we hope to solve a DLP more quickly.

VIII.2. The GHS Attack

VIII.2.1. The Reduction. The most important case for practical applications are elliptic curves in characteristic two. In this section we describe the reduction of a DLP on such an elliptic curve over K to the divisor class group of a curve defined over k .

We start with an ordinary elliptic curve

$$\mathcal{E}_{a,b} : Y^2 + XY = X^3 + aX^2 + b \quad (\text{VIII.1})$$

with $a, b \in K$, $b \neq 0$. Every isomorphism class of ordinary elliptic curves over K has a unique representative of the form (VIII.1) under the requirement that $a \in \{0, \omega\}$ where $\omega \in \mathbb{F}_{2^u}$ with $u = 2^{v_2(nr)}$ is a fixed element such that $\text{Tr}_{\mathbb{F}_{2^u}/\mathbb{F}_2}(\omega) = 1$. In the following we only consider these unique elliptic curves with $a \in \{0, \omega\}$.

Applying the transformations $Y = y/\tilde{x} + b^{1/2}$, $X = 1/\tilde{x}$, multiplying by $\tilde{x}^2$, substituting $\tilde{x} = x/\gamma$ for some $\gamma \in K^\times$ and writing $\alpha = a$, $\beta = b^{1/2}/\gamma$ we obtain the Artin-Schreier equation

$$y^2 + y = \gamma/x + \alpha + \beta x. \quad (\text{VIII.2})$$

On the other hand, reversing the transformations, we can return to equation (VIII.1) from equation (VIII.2) for any $\gamma, \beta \in K^\times$. This shows that the function field $E = K(\mathcal{E}_{a,b})$ of $\mathcal{E}_{a,b}$ contains and is in fact generated by two functions x, y satisfying the relation (VIII.2), that is, $E = K(x, y)$. Note that the transformation backwards is described by the map $(\gamma, \beta) \mapsto b = (\gamma\beta)^2$ and $a = \alpha$.

The function field $E = K(x, y)$ is a Galois extension of the rational function field $K(x)$ of degree two. Furthermore, $K(x)$ is a Galois extension of $k(x)$ of degree n and the Galois group is generated by the Frobenius automorphism σ of $K(x)$ with respect to K/k satisfying $\sigma(x) = x$. We define the function field $C = C_{\gamma, \alpha, \beta}$ to be the splitting field of the extension $E/k(x)$.

Before we state the main theorem about $C_{\gamma, \alpha, \beta}$ we need some further notation. For $z \in K$ let $m_z(t) = \sum_{i=0}^m \lambda_i t^i \in \mathbb{F}_2[t]$ with $\lambda_m = 1$ be the unique polynomial of minimal degree such that $\sum_{i=0}^m \lambda_i \sigma^i(z) = 0$. We define $m_{\gamma, \beta} = \text{lcm}\{m_\gamma, m_\beta\}$.

Recall that if we have a Frobenius automorphism on $C_{\gamma, \alpha, \beta}$ with respect to K/k , we take $C^0 = C_{\gamma, \alpha, \beta}^0$ to be its fixed field.

THEOREM VIII.1. *The Frobenius automorphism σ of $K(x)$ with respect to K/k satisfying $\sigma(x) = x$ extends to a Frobenius automorphism of C with respect to K/k if and only if at least one of the conditions*

$$\text{Tr}_{K/\mathbb{F}_2}(\alpha) = 0, \text{Tr}_{K/k}(\gamma) \neq 0 \text{ or } \text{Tr}_{K/k}(\beta) \neq 0$$

holds. In this case, we have:

(i) *If $k(\gamma, \beta) = K$, then*

$$\ker(\phi) \subseteq \mathcal{E}(K)[2^{\deg(m_{\gamma, \beta})-1}].$$

(ii) *The genus of C^0 satisfies*

$$g_{C^0} = 2^{\deg(m_{\gamma, \beta})} - 2^{\deg(m_{\gamma, \beta}) - \deg(m_\gamma) - \deg(m_\beta)} + 1.$$

(iii) *There is a rational subfield of C^0 of index $\min\{2^{\deg(m_\gamma)}, 2^{\deg(m_\beta)}\}$.*

(iv) *If $\gamma \in k$ or $\beta \in k$, then C^0 is hyperelliptic.*

For the proof of Theorem VIII.1 see [145], [168] and Section VIII.2.6. The following corollary is an immediate consequence of Theorem VIII.1 and

the fact that m_β is not divisible by $t-1$ if and only if the given trace condition holds. A proof is given in [168].

COROLLARY VIII.2. *If $\gamma \in k$, then*

$$g_{C^0} = \begin{cases} 2^{\deg(m_{\gamma,\beta})-1} - 1 & \text{if } \text{Tr}_{K/\mathbb{F}_{q^u}}(\beta) = 0 \text{ where } u = 2^{v_2(n)}, \\ 2^{\deg(m_{\gamma,\beta})-1} & \text{otherwise.} \end{cases}$$

Similarly with γ and β exchanged.

The construction of C^0 and the computation of images under ϕ can be made explicit using Artin-Schreier extensions and the operation of σ on C ; see [145], [168] and [165]. This leads to algorithms which are at most polynomial in g_{C^0} . Various implementations of this construction are available in the computer algebra systems *Kash* and *Magma* [46], [193], [229].

If the condition of Theorem VIII.1(i) is satisfied, we see that any large prime factor of $\mathcal{E}(K)$ and hence the DLP in $\mathcal{E}(K)$ is preserved under ϕ . Since there is some freedom in choosing γ and β , this can also be achieved if $\mathcal{E}$ is actually defined over a subfield of K . We also remark that there is an explicit formula for the L -polynomial (or zeta function) of C^0 in terms of the L -polynomials (or zeta functions) of $\mathcal{E}$ and further related elliptic curves; see [165], [168] and Section VIII.5.3.

The main points of interest are the possible degrees $m = \deg(m_{\gamma,\beta})$ and the relationship to γ and β . The efficiency or feasibility of the attack crucially depends on this m , and it is therefore sometimes referred to as the “magic” number m . Its properties are discussed in Section VIII.2.4.

We conclude this section with some examples.

Example 1: Let $k = \mathbb{F}_2$ and $K = \mathbb{F}_{2^{127}} = k[w]$ with $w^{127} + w + 1 = 0$. Consider the elliptic curve

$$\mathcal{E} : Y^2 + XY = X^3 + w^2.$$

We can choose $\gamma = 1$, $\alpha = 0$ and $\beta = w$. Then $m_\gamma(t) = t + 1$ and $m_\beta(t) = t^7 + t + 1$ since $\beta^{128} + \beta^2 + \beta = 0$. It follows that $m_{\gamma,\beta}(t) = t^8 + t^7 + t^2 + 1$. All conditions of Theorem VIII.1 are fulfilled. We conclude that C^0 is a hyperelliptic function field over k of genus 127. Using the programs mentioned above we compute a representing curve

$$C^0 : y^2 + (x^{127} + x^{64} + 1)y = x^{255} + x^{192} + x^{128}.$$

A different model is given by $C^0 : y^2 + (x^{128} + x^{64} + x)y = (x^{128} + x^{64} + x)$.

Example 2: Let $k = \mathbb{F}_{2^5} = \mathbb{F}_2[u]$ and $K = \mathbb{F}_{2^{155}} = \mathbb{F}_2[w]$ with $u^5 + u^2 + 1 = 0$ and $w^{155} + w^{62} + 1 = 0$. Consider the elliptic curve

$$\mathcal{E} : Y^2 + XY = X^3 + \delta$$

with

$$\delta = \begin{aligned} &w^{140} + w^{134} + w^{133} + w^{132} + w^{130} + w^{129} + w^{128} + w^{127} + w^{117} + \\ &w^{113} + w^{111} + w^{110} + w^{102} + w^{97} + w^{96} + w^{78} + w^{74} + w^{72} + \\ &w^{70} + w^{63} + w^{49} + w^{48} + w^{47} + w^{41} + w^{39} + w^{36} + w^{35} + w^{34} + \\ &w^{32} + w^{24} + w^{17} + w^{10} + w^9 + w^8 + w^5. \end{aligned}$$

Similarly to the previous example we can choose $\gamma = 1$, $\alpha = 0$ and $\beta = \delta^{1/2}$. Then $m_\gamma(t) = t + 1$ and $m_\beta(t) = t^{15} + t^{11} + t^{10} + t^9 + t^8 + t^7 + t^5 + t^3 + t^2 + t + 1$. It follows that $m_{\gamma,\beta}(t) = m_\gamma(t)m_\beta(t)$. All conditions of Theorem VIII.1 are fulfilled and we conclude that C^0 is a hyperelliptic function field over k of genus 32767.

Example 3: In Example 2 we can also choose

$$\gamma = \frac{w^{140} + w^{132} + w^{128} + w^{125} + w^{101} + w^{94} + w^{78} + w^{70} + w^{64} + w^{63} + w^{47} + w^{39} + w^{35}}{w^{78} + w^{70} + w^{64} + w^{63} + w^{47} + w^{39} + w^{35}},$$

$\alpha = 0$ and $\beta = \delta^{1/2}/\gamma$. Then $m_{\gamma,\beta}(t) = m_\gamma(t) = m_\beta(t) = t^5 + t^2 + 1$. All conditions of Theorem VIII.1 are fulfilled and we conclude that C^0 is a function field over k of genus 31. Using the programs mentioned above we compute a representing curve

$$\mathcal{C}^0 : \begin{aligned} &y^{32} + u^{22}y^{16} + u^3y^8 + u^9y^4 + u^{13}y^2 + u^{24}y + (u^{24}x^{24} + u^9x^{16} \\ &+ u^{25}x^{12} + u^{30}x^{10} + u^3x^9 + u^{26}x^7 + u^{23}x^6 + u^{15}x^4 + u^{30})/x^8 = 0. \end{aligned}$$

The last two examples show that the choice of γ and β can make a very significant difference for the size of the resulting genus.

VIII.2.2. The Asymptotic Attack. Let us assume that $\gamma \in k$. According to Theorem VIII.1(iv) and Corollary VIII.2, the resulting function field C^0 is hyperelliptic and has genus bounded by $2^{n-1} - 1$ or 2^{n-1} since $m_{\gamma,\beta}(t)$ divides $t^n - 1$, and this holds independently of q . Combining this with the theorem of Gaudry, Theorem VII.10, and the improvements of Harley for very small genera we obtain the following, slightly improved main result of [145].

THEOREM VIII.3. *Let $\mathcal{E} : Y^2 + XY = X^3 + \alpha X^2 + \beta$ denote an elliptic curve over $K = \mathbb{F}_{q^n}$ such that*

$$n \text{ is odd or } \text{Tr}_{K/\mathbb{F}_2}(\alpha) = 0 \text{ or } \text{Tr}_{K/k}(\beta) \neq 0.$$

Let $\#\mathcal{E}(K) = \ell h$, where ℓ is a large prime. One can solve the discrete logarithm problem in the ℓ -cyclic subgroup of $\mathcal{E}(K)$ in time $O(q^2)$ where the complexity estimate holds for a fixed value of $n \geq 4$ as q tends to infinity.

The complexity estimate is in fact always slightly better than $O(q^2)$. This time has to be compared with the running time of the Pollard methods on $\mathcal{E}(K)$, which are $O(q^{n/2})$. It follows that the DLP can be solved (much) faster using the GHS attack when $n \geq 4$ is fixed and q tends to infinity. By the

recent results of Thériault [325] improving index-calculus for hyperelliptic curves further we find that the DLP on C^0 can be solved in time $O(q^{10/7})$ and $O(q^{14/9})$ if the genus is 3 or 4, respectively. Since $10/7 < 3/2 < 14/9$, this means that the DLP on $\mathcal{E}$ can also be solved asymptotically faster using the GHS attack when $n = 3$ and $\text{Tr}_{K/k}(\beta) = 0$, using Corollary VIII.2.

It is now natural to ask, at what sizes of q does the crossover point lie. A computer experiment comparing the Pollard times against index-calculus times for $n = 4$ has been carried out in [145]. The index-calculus method proves to be faster by a factor of about 0.7 for an 84-bit elliptic curve. Since the crossover point is already reached for such small field sizes, it can be concluded that for $n = 4$ the DLP is solved faster using the GHS attack also in practical instances and not only asymptotically. The crossover point for larger values of n however will be much higher (see for example [311]).

VIII.2.3. Special Attacks. The previous section applies uniformly to all elliptic curves when n is small and q large enough, so that g_{C^0} is relatively small. But it could also apply for cases where n is large and $\deg(m_{\gamma,\beta})$ happens to be sufficiently small. More generally, we could allow arbitrary $\gamma \in K$, resulting in non-hyperelliptic curves C^0 , and consider only those cases where g_{C^0} is sufficiently small. Moreover, the running time of the algorithm behind Theorem VIII.3 depends exponentially on g_{C^0} , since low genus index-calculus methods are used. But there are also high genus index-calculus methods available, see [92], [117] and [169], whose asymptotic running time depends subexponentially on g_{C^0} and is roughly of the form $q^{g_{C^0}^{1/2+o(1)}}$. In Example 1 we have $g_{C^0} = n$, and for such cases with large n the high genus index-calculus attack would be much more efficient than the Pollard methods on the original elliptic curve.

In the following sections we investigate for which special values of n and which special families of elliptic curves an index-calculus attack can be mounted which is (possibly) faster than the Pollard methods. A summary of the results of practical interest also using the methods of Section VIII.3 is given in Section VIII.4.

VIII.2.4. Analysis of Possible Genera and Number of Curves. Possible genera of C^0 and the number of corresponding elliptic curves have been investigated in detail in [233], [241] for the case $\gamma \in k$. We give here a more general and simplified discussion allowing for arbitrary $\gamma \in K$.

We first analyse the various possibilities for $m_\beta(t)$ and β in more detail. It is helpful to introduce the following standard technique from linear algebra (see also [134] and [241]). We define a multiplication of polynomials $h(t) = \sum_{i=0}^d \lambda_i t^i \in k[t]$ and finite field elements $z \in K$ by $h(t)z = \sum_{i=0}^d \lambda_i \sigma^i(z)$. This makes the additive group of K into a so-called $k[t]$ -module. The polynomials $m_\beta(t)$ are then by definition polynomials in $\mathbb{F}_2[t]$ of smallest degree such that $m_\beta(t)\beta = 0$.

Let $w \in K$ be a normal basis element for K/k . The theorem of the normal basis is equivalent to the statement that K is a cyclic (or “one-dimensional”) $k[t]$ -module with annihilator $t^n - 1$, that is, $K = \{h(t)w : h(t) \in k[t] \text{ and } \deg(h(t)) < n\} \cong k[t]/(t^n - 1)$ is generated by one element. We define $B_{m(t)} = \{\gamma \in K : m_\gamma(t) = m(t)\}$ for $m(t) \in k[t]$ and let $\Phi(m(t))$ be the number of polynomials of degree less than n in $k[t]$ coprime to $m(t)$. These observations and definitions give the following theorem.

THEOREM VIII.4. *For every $\beta \in K$ it holds that $m_\beta(t)$ is a divisor of $t^n - 1$ in $\mathbb{F}_2[t]$. Conversely, let $m(t)$ be a divisor of $t^n - 1$ in $\mathbb{F}_2[t]$. Then*

$$B_{m(t)} = \{h(t)\gamma_0 : h \in k[t], \deg(h) < \deg(m) \text{ and } \gcd\{h, m\} = 1\}$$

with $\gamma_0 = ((t^n - 1)/m(t))w$ and

$$\#B_{m(t)} = \Phi(m(t)) = \prod_{i=0}^s (q^{j_i d_i} - q^{(j_i - 1)d_i}),$$

where $m(t) = \prod_{i=0}^s f_i^{j_i}$ is the factorization of $m(t)$ into irreducible polynomials $f_i \in \mathbb{F}_2[t]$ with $\deg(f_i) = d_i$.

We remark that the situation is completely analogous to the possibly more familiar case of finite cyclic groups. Consider an additive cyclic group G with generator g of order M . Replace K by G , w by g , $k[t]$ by $\mathbb{Z}$, $t^n - 1$ by M and $k[t]/(t^n - 1)$ by $\mathbb{Z}/(M)$ and let m be a divisor of M . The analogous version of Theorem VIII.4 then tells us, for example, that the elements $\gamma \in G$ of precise order $m_\gamma = m$ are given in the form $h\gamma_0$ with $0 \leq h < m$, $\gcd\{h, m\} = 1$ and $\gamma_0 = (M/m)g$.

Using Theorem VIII.4, all possible $m_\beta(t)$, the corresponding β and their cardinalities can be easily computed. Combining the various possibilities for $m_\gamma(t)$ and $m_\beta(t)$ yields all possible $m_{\gamma,\beta}(t)$ and genera g_{C^0} . But we also require that $k(\gamma, \beta) = K$, so not all combinations do actually occur. We can obtain sharp lower bounds for g_{C^0} as follows. Let $n_1 = [k(\gamma) : k]$ and $n_2 = [k(\beta) : k]$. Then $n = \text{lcm}\{n_1, n_2\}$. Furthermore, $m_\gamma(t)$ divides $t^{n_1} - 1$ but does not divide $t^s - 1$ for any proper divisor s of n_1 , and analogously for $m_\beta(t)$. Enumerating the smallest possibilities for $m_\gamma(t)$ and $m_\beta(t)$ for all n_1, n_2 then yields sharp lower bounds.

The following theorem contains some statistics about the possible genera.

THEOREM VIII.5. *Assume that the trace condition of Theorem VIII.1 and $k(\gamma, \beta) = K$ holds. Then $n \leq g_{C^0} \leq 2^n - 1$ and $g_{C^0} \geq 65535$ for all primes $100 \leq n \leq 1000$ except $g_{C^0} = 127$ for $n = 127$ and $g_{C^0} = 32767$ for $n = 151$. The lower bound is attained if and only if*

$$n \in \{1, 2, 3, 4, 7, 15, 21, 31, 63, 93, 105, 127, 217, 255, 381, 465, 511, 889\}.$$

Among these, the values $n \in \{1, 2, 3, 4, 7, 15, 31, 63, 127, 255, 511\}$ yield an (elliptic or) hyperelliptic function field C^0 .

The proof of Theorem VIII.5 follows the above observations and requires explicit calculations using a computer. We remark that $n \leq g_{C^0} \leq 2n$ for 43 odd and even values of $n \in \{1, \dots, 1000\}$.

Theorem VIII.5 basically means that the GHS attack in even characteristic fails for large prime values of n since it does not appear that the DLP can be solved more easily in a curve of genus ≥ 65535 , albeit defined over k instead of K . On the other hand, composite values of n and in particular the special values given in the list appear susceptible. Note that the extension degree 155 of Example 3 is not shown in Theorem VIII.5; however, $n = 31$ is a factor of 155 which was indeed used to construct a curve of genus 31.

Let us now look at the proportion of elliptic curves which yield a small g_{C^0} when n is large. In view of the running time for high genus index-calculus methods, a rough estimation of possibly interesting genera is $g_{C^0} \leq n^2$. The following lemma contains a crude upper bound for the proportion of possibly susceptible elliptic curves.

LEMMA VIII.6. *Let $\rho \geq 1$. The probability that a C^0 associated to a random elliptic curve $\mathcal{E}_{a,b}$ has a genus at most n^ρ is bounded by approximately $2^{2\rho \log_2(n)+2} q^{2\rho \log_2(n)} / q^n$.*

PROOF. It is not difficult to see that $2^{m-d_1} + 2^{m-d_2} \leq 2^{m-1} + 2$ under the side conditions $1 \leq d_1, d_2 \leq m$, $d_1 + d_2 \geq m$. Thus with $d_1 = \deg(m_\gamma)$, $d_2 = \deg(m_\beta)$ and Theorem VIII.1(ii) it follows that $n^2 \geq g_{C^0} \geq 2^{m-1} - 1$ and $m \leq \log_2(n^\rho + 1) + 1$. Now there are at most 2^{m+1} polynomials over $\mathbb{F}_2$ of degree $\leq m$, so there are at most 2^{2m+2} pairs (m_γ, m_β) such that $m_{\gamma,\beta} = \text{lcm}\{m_\gamma, m_\beta\}$ has degree $\leq m$. Consequently there are at most $2^{2m+2} q^{2m}$ pairs (γ, β) and thus elements $b = (\gamma\beta)^2$. The total number of b is q^n , so the result follows with $m \approx \rho \log_2(n)$. $\square$

As a result we see that the probability of obtaining a relatively small genus for a random elliptic curve quickly becomes negligible as n increases.

The following results are additions to Theorem VIII.4 and can be found in [241].

LEMMA VIII.7. *Let n be an odd prime. The polynomial $t^n - 1$ factors over $\mathbb{F}_2$ as $t^n - 1 = (t - 1)\psi_n(t) = (t - 1)h_1(t) \cdots h_s(t)$, where $\psi_n(t)$ denotes the n th cyclotomic polynomial and the $h_i(t)$ are distinct polynomials of a degree d . Furthermore, d is the order of 2 in $(\mathbb{Z}/n\mathbb{Z})^\times$ and $d \geq \log_2(n + 1)$.*

COROLLARY VIII.8. *Let $\delta \in \{0, 1\}$. For any $\beta \in K$, the degree of $m_\beta(t)$ is of the form $\deg(m_\beta(t)) = id + \delta$, and there are $\binom{s}{i} (q^d - 1)^i (q - 1)^\delta$ different $\beta \in K$ such that $\deg(m_\beta(t)) = id + \delta$.*

The corollary follows immediately from Theorem VIII.4 and Lemma VIII.7.

VIII.2.5. Further Analysis and the Decompositions $\mathbf{b} = \gamma\beta$. In the following let $m_1(t), m_2(t) \in \mathbb{F}_2[t]$ denote divisors of $t^n - 1$ such that if $t^s - 1$

is divisible by $m_1(t)$ and by $m_2(t)$, then s is divisible by n . In other words, we require $K = k(\gamma, \beta)$ for every $\gamma \in B_{m_1(t)}$ and $\beta \in B_{m_2(t)}$. We abbreviate this condition on m_1 and m_2 by the predicate $P(m_1, m_2)$ and define

$$B_{m_1(t), m_2(t)} = \{\gamma\beta : \gamma \in B_{m_1(t)}, \beta \in B_{m_2(t)}\}.$$

We remark that $B_{m_1(t), m_2(t)}$ is invariant under the 2-power Frobenius automorphism. To distinguish cases we define the predicate

$$T(m_1(t), m_2(t)) = (v_{t-1}(m_1(t)) = 2^{v_2(n)} \text{ or } v_{t-1}(m_2(t)) = 2^{v_2(n)}).$$

Then, let

$$S_{m_1(t), m_2(t)} = \begin{cases} \{\mathcal{E}_{a,b} : a \in \{0, \omega\}, b \in B_{m_1(t), m_2(t)}\} & \text{if } T(m_1(t), m_2(t)), \\ \{\mathcal{E}_{0,b} : b \in B_{m_1(t), m_2(t)}\} & \text{otherwise.} \end{cases}$$

Observe that $T(m_1(t), m_2(t))$ holds true precisely when $\text{Tr}_{K/k}(\gamma) \neq 0$ or $\text{Tr}_{K/k}(\beta) \neq 0$. Thus by Theorem VIII.1 and since $P(m_1, m_2)$ is assumed to hold true, the set $S_{m_1(t), m_2(t)}$ contains elliptic curves for which the GHS reduction applies when using the corresponding γ, α, β . Letting $m_{\gamma, \beta}(t) = \text{lcm}\{m_1(t), m_2(t)\}$ and $m = \deg(m_{\gamma, \beta}(t))$, the resulting genus satisfies $g_{C^0} = 2^m - 2^{m-\deg(m_1(t))} - 2^{m-\deg(m_2(t))} + 1$.

We say that an elliptic curve $\mathcal{E}_{a,b}$ is susceptible to the GHS attack if $\mathcal{E}_{a,b} \in S_{m_1(t), m_2(t)}$ for some “suitable” choices of $m_1(t), m_2(t)$. Here “suitable” means that $m_1(t), m_2(t)$ are such that we expect that the DLP can be solved more easily in $\text{Pic}_k(C^0)$ than by the Pollard methods in $\mathcal{E}_{a,b}(K)$. With regard to Section VIII.2.3, the main questions then are how to find such suitable choices of $m_1(t), m_2(t)$, how to determine the cardinality of $S_{m_1(t), m_2(t)}$ and how to develop an efficient algorithm which checks whether $\mathcal{E}_{a,b} \in S_{m_1(t), m_2(t)}$ and computes the corresponding decomposition $b = \gamma\beta$ for $\gamma, \beta \in K$ with $m_\gamma = m_1$ and $m_\beta = m_2$. The following lemma answers these questions in the case of a composite n .

LEMMA VIII.9. *Let $n = n_1 n_2$, $K_1 = \mathbb{F}_{q^{n_1}}$ and $b \in K$. Let $f_1 = t^{n_1} - 1$ and $f_2 = (t - 1)(t^n - 1)/(t^{n_1} - 1)$.*

(i) *The following conditions are equivalent.*

1. $\text{Tr}_{K/K_1}(b) \neq 0$.
2. *There exist $\gamma_1, \gamma_2 \in K^\times$ with*

$$\gamma_1 \gamma_2 = b, \gamma_1 \in K_1 \text{ and } \text{Tr}_{K/K_1}(\gamma_2) \in k^\times.$$

3. *There exist $\gamma_1, \gamma_2 \in K^\times$ with*

$$\gamma_1 \gamma_2 = b, m_{\gamma_1} \text{ dividing } f_1, m_{\gamma_2} \text{ dividing } f_2 \text{ and } v_{t-1}(m_{\gamma_2}) = v_{t-1}(f_2).$$

(ii) *If $\text{Tr}_{K/K_1}(b) \neq 0$, then $\gamma_1 = \text{Tr}_{K/K_1}(b)$ and $\gamma_2 = b/\gamma_1$ satisfy conditions 2 and 3 of (i). For any two decompositions $b = \gamma_1 \gamma_2 = \tilde{\gamma}_1 \tilde{\gamma}_2$ as in (i) it holds that $\gamma_1/\tilde{\gamma}_1 \in k^\times$.*

(iii) Let m_1 divide f_1 and m_2 divide f_2 such that $v_{t-1}(m_2) = v_{t-1}(f_2)$. Then

$$\#B_{m_1, m_2} = \Phi(m_1)\Phi(m_2)/(q-1).$$

(iv) (GHS conditions). In the case of (ii):

1. $k(b) = k(\gamma_1, \gamma_2)$,
2. $\text{Tr}_{K/k}(\gamma_2) \neq 0$ if and only if n_1 is odd,
3. $\text{Tr}_{K/k}(\gamma_1) \neq 0$ if and only if $v_{t-1}(m_b) = 2^{v_2(n)}$ and n/n_1 is odd.

PROOF.

(i)

- 1 $\Rightarrow$ 2. Define $\gamma_1 = \text{Tr}_{K/K_1}(b)$ and $\gamma_2 = b/\gamma_1$, observing $\gamma_1 \neq 0$ by assumption. Then $\text{Tr}_{K/K_1}(\gamma_2) = \text{Tr}_{K/K_1}(b)/\gamma_1 = 1$, which proves the first implication.
- 2 $\Rightarrow$ 3. If $\gamma_1 \in K_1$, then $(t^{n_1} - 1)\gamma_1 = 0$ and hence m_{γ_1} divides f_1 . Also, $\text{Tr}_{K/K_1}(\gamma_2) = ((t^n - 1)/(t^{n_1} - 1))\gamma_2 \neq 0$ and $(t-1)\text{Tr}_{K/K_1}(\gamma_2) = 0$ since $\text{Tr}_{K/K_1}(\gamma_2) \in k^\times$. This implies $v_{t-1}(m_{\gamma_2}) = v_{t-1}(f_2)$ and m_{γ_2} divides f_2 , and the second implication follows.
- 3 $\Rightarrow$ 1. Since m_{γ_1} divides f_1 we have $\gamma_1 \in K_1$. The conditions $v_{t-1}(m_{\gamma_2}) = v_{t-1}(f_2)$ and m_2 divides f_2 imply that

$$\text{Tr}_{K/K_1}(\gamma_2) = ((t^n - 1)/(t^{n_1} - 1))\gamma_2 \neq 0.$$

Also $\gamma_1 \neq 0$ by assumption, so that $\text{Tr}_{K/K_1}(b) = \gamma_1 \text{Tr}_{K/K_1}(\gamma_2) \neq 0$. This proves the third implication.

(ii) The first part follows from the proof of (i). Let $b = \gamma_1\beta_1 = \gamma_2\beta_2$ be the two decompositions where $\gamma_i \in K_1$ and $\mu_i = \text{Tr}_{K/K_1}(\beta_i) \in k^\times$ by (i). Then $\gamma_1\mu_1 = \text{Tr}_{K/K_1}(b) = \gamma_2\mu_2 \neq 0$. Thus $\gamma_1/\gamma_2 \in k^\times$.

(iii) Using Theorem VIII.4 and observing $\gamma_1\gamma_2 = (\lambda\gamma_1)(\lambda^{-1}\gamma_2)$ for $\lambda \in k^\times$, it follows that $\#B_{m_1, m_2} \leq \Phi(m_1)\Phi(m_2)/(q-1)$. Because of (ii) this is in fact an equality.

(iv)

1. $k(b) \subseteq k(\gamma_1, \gamma_2)$ is clear since $b = \gamma_1\gamma_2$. On the other hand, $\gamma_2 = \text{Tr}_{K/K_1}(b) \in k(b)$, hence $\gamma_1 = b/\gamma_2 \in k(b)$ and $k(\gamma_1, \gamma_2) \subseteq k(b)$ follows.
2. Writing $\lambda = \text{Tr}_{K/K_1}(\gamma_2)$ we have $\text{Tr}_{K/k}(\gamma_2) = \text{Tr}_{K_1/k}(\lambda) = n_1\lambda$ since $\lambda \in k^\times$.
3. We have that $\text{Tr}_{K/k}(\gamma_1) \neq 0$ is equivalent to $v_{t-1}(m_{\gamma_1}) = 2^{v_2(n)}$. Also, we have m_{γ_1} divides m_b since $\gamma_1 = ((t^n - 1)/(t^{n_1} - 1))b$ and $v_{t-1}(m_{\gamma_1}) = v_{t-1}(m_b)$ if and only if n/n_1 is odd. This implies the statement.

□

COROLLARY VIII.10. *Assume that n_1 is odd. Then*

$$\left\{ \mathcal{E}_{a,b} : \begin{array}{l} a \in \{0, \omega\}, \\ \text{Tr}_{K/K_1}(b) \neq 0, \\ k(b) = K \end{array} \right\} = \bigcup \left\{ S_{m_1, m_2} : \begin{array}{l} m_1 \text{ divides } f_1, \\ m_2 \text{ divides } f_2, \\ v_{t-1}(m_2) = 2^{v_2(n)}, \\ P(m_1, m_2) \end{array} \right\},$$

where the union is disjoint.

Example 4: Consider $n = 6$, $n_1 = 3$ and $n_2 = 2$. Using Corollary VIII.10 we see that every elliptic curve over $\mathbb{F}_{q^6}$ with $\text{Tr}_{\mathbb{F}_{q^6}/\mathbb{F}_{q^3}}(b) \neq 0$ leads to a genus $g_{C^0} \in \{8, 9, 11, 12, 14\}$.

We now focus on the case where n is an odd prime. We can then use Lemma VIII.7 and Corollary VIII.8. Over $\mathbb{F}_2$ we have the factorization into irreducible polynomials $t^n + 1 = (t - 1)h_1 \cdots h_s$ and $\deg(h_i) = d$ such that $n = sd + 1$. In this situation the first non-trivial $m = \deg(m_{\gamma, \beta})$ satisfies $d \leq m \leq d + 1$ corresponding to $m_{\gamma, \beta} = h_i$ or $m_{\gamma, \beta} = (t - 1)h_i$ by Theorem VIII.4 and equation (VIII.5), observing that $(t - 1) \nmid h_i$. After that we already have $m \geq 2d$ which is too big in most instances. The case $m = d$ is (only) obtained when $\gamma, \beta \in B_{h_i(t)}$ and $\text{Tr}_{K/\mathbb{F}_2}(\alpha) = 0$. The conditions of Theorem VIII.1 are fulfilled so the GHS reduction does work and the resulting genus is $g_{C^0} = 2^d - 1$. The case $m = d + 1$ leads to the smallest genera when $\gamma \in B_{t-1}$ and $\beta \in B_{h_i(t)} \cup B_{(t-1)h_i(t)}$, which is a disjoint union. Since $\text{Tr}_{K/k}(\gamma) = n\gamma \neq 0$ the conditions of Theorem VIII.1 are fulfilled for every $\alpha \in \{0, \omega\}$, and the genus g_{C^0} is $2^d - 1$ if $\beta \in B_{h_i(t)}$ and 2^d if $\beta \in B_{(t-1)h_i(t)}$.

The transformation between (VIII.2) and (VIII.1) is described by $b = (\gamma\beta)^2$ and $a = \alpha$. The map $(\gamma, \beta) \mapsto (\gamma\beta)^2$ is not injective, so several tuples (γ, β) will lead to the same elliptic curve. It is $(q - 1) - 1$ when restricted to $B_{t-1} \times B_{h_i}$ or $B_{t-1} \times B_{(t-1)h_i}$, and at least $2(q - 1) - 1$ when restricted to $B_{h_i} \times B_{h_i}$. Namely, the tuples $(\lambda\gamma, \lambda^{-1}\beta)$ for $\lambda \in k^\times$ and also (β, γ) in the latter case are mapped to the same b as (γ, β) . Heuristically we expect that the fibres are not much bigger than $2(q - 1)$ in this case if $\#(B_{h_i} \times B_{h_i})$ is small in comparison with $\#K$. Now clearly $B_{t-1, h_i} = B_{h_i}$ and $B_{t-1, (t-1)h_i} = B_{(t-1)h_i}$, so these sets are disjoint for all i and we obtain $\# \cup_i B_{t-1, h_i} = s(q^d - 1)$ and $\# \cup_i B_{t-1, (t-1)h_i} = s(q - 1)(q^d - 1)$. Furthermore, we expect that $\#B_{h_i, h_i} \approx \min\{q^n, (q^d - 1)^2 / (2(q - 1))\}$ and that $\# \cup_i B_{h_i, h_i} \approx \min\{q^n, s(q^d - 1)^2 / (2(q - 1))\}$.

A summary is given in Table VIII.1. Note that the elliptic curves from S_{m_γ, m_β} of the last two rows lead to hyperelliptic function fields C^0 and that the cardinality † is only heuristically expected (Lemma VIII.11 provides proven bounds in special cases).

If h_i is of a trinomial form, we have the following precise statement.

TABLE VIII.1. Cardinalities of S_{m_γ, m_β} for n Odd Prime

m_γ	m_β	α	g_{C^0}	$\approx \#S_{m_\gamma, m_\beta}$
h_i	h_i	0	$2^d - 1$	$\min\{q^n, sq^{2d-1}/2\}^\dagger$
$t-1$	h_i	$0, \omega$	$2^d - 1$	$2sq^d$
$t-1$	$(t-1)h_i$	$0, \omega$	2^d	$2sq^{d+1}$

LEMMA VIII.11. Assume $h = t^{r_1} + t^{r_2} + 1$ with $r_1 > r_2 > 0$, $\gcd\{r_2, n\} = 1$ and $m_b \nmid h$. Then there are no or precisely $2(q-1)$ pairs $(\gamma, \beta) \in K^2$ such that $b = \gamma\beta$ and $m_\gamma = m_\beta = h$.

PROOF. Assume that $b = \gamma\beta$ and $m_\gamma = m_\beta = h$. Then $\gamma \neq 0$ and $\beta \neq 0$ since $b \neq 0$, and consequently

$$\begin{aligned}\gamma^{q^{r_1}-1} + \gamma^{q^{r_2}-1} + 1 &= 0, \\ b^{q^{r_1}-1} + b^{q^{r_2}-1}\gamma^{q^{r_1}-q^{r_2}} + \gamma^{q^{r_1}-1} &= 0.\end{aligned}$$

Define $\rho = \gamma^{q^{r_2}-1}$. The first equation implies

$$\begin{aligned}\gamma^{q^{r_1}-1} &= \rho + 1, \\ \gamma^{q^{r_1}-q^{r_2}} &= (\rho + 1)/\rho.\end{aligned}\tag{VIII.3}$$

Substituting this into the second equation yields $b^{q^{r_1}-1} + b^{q^{r_2}-1}(\rho + 1)/\rho + \rho + 1 = 0$, and then

$$\rho^2 + (b^{q^{r_1}-1} + b^{q^{r_2}-1} + 1)\rho + b^{q^{r_2}-1} = 0.\tag{VIII.4}$$

On the other hand, any further solution ρ, γ, β to (VIII.4), (VIII.3), $\gamma^{q^{r_2}-1} = \rho$ and $\beta = b/\gamma$ satisfies $m_\gamma = m_\beta = h$ because $b \neq 0$.

Since $m_b \nmid h$ we have that $\gamma/\beta \notin k$. There are thus at least $2(q-1)$ pairwise distinct solutions of the form $(\lambda\gamma, \lambda^{-1}\beta)$ and $(\lambda\beta, \lambda^{-1}\gamma)$ with $\lambda \in k^\times$. On the other hand, there are at most two possibilities for ρ and at most $q-1$ possibilities for γ for each ρ , resulting in at most $2(q-1)$ solutions. Namely, for any two solutions γ_1, γ_2 with $\gamma_i^{q^{r_2}-1} = \rho$ we have that $(\gamma_1/\gamma_2)^{q^{r_2}-1} = 1$ and hence $\gamma_1/\gamma_2 \in \mathbb{F}_{q^{r_2}} \cap \mathbb{F}_{q^n}^\times = \mathbb{F}_q^\times$ since r_2 and n are coprime. $\square$

Given b and h as in the lemma, γ and β can be computed efficiently as follows, using Langrange's resolvent [210, p. 289]:

1. Solve for ρ such that $\rho^2 + (b^{q^{r_1}-1} + b^{q^{r_2}-1} + 1)\rho + b^{q^{r_2}-1} = 0$.
2. Compute θ such that

$$\gamma = \theta + \rho^{-1}\theta^{q_2} + \rho^{-1-q_2}\theta^{q_2^2} + \dots + \rho^{-1-q_2-\dots-q_2^{n-2}}\theta^{q_2^{n-1}} \neq 0,$$

where $q_2 = q^{r_2}$. This can be achieved by linear algebra over k .

3. Compute $\beta = b/\gamma$.

If $N_{K/k}(\rho) \neq 1$ in step 1 or if γ does not satisfy (VIII.3) in step 2, then there are no solutions γ, β with $m_\gamma = m_\beta = h$.

Example 5: Consider $n = 7$. Then $t^n - 1 = (t - 1)(t^3 + t + 1)(t^3 + t^2 + 1)$, $d = 3$ and $s = 2$. Using the first row of Table VIII.1 we see that a proportion of about q^{-2} of all elliptic curves over $\mathbb{F}_{q^7}$ with $\alpha = 0$ leads to $g_{C^0} = 7$.

In Lemmas VIII.9 and VIII.11 we have discussed the decomposition $b = \gamma\beta$ and how to check $\mathcal{E}_{a,b} \in S_{m_1, m_2}$ in some special cases. A simple and the currently only known method to do this in full generality is to take all $\gamma \in B_{m_1}$ and to test whether $m_{b/\gamma} = m_2$. Of course, for any γ we do not need to check $\lambda\gamma$ with $\lambda \in k^\times$.

VIII.2.6. Further Details. In this section we present some details on the Artin–Schreier construction of Theorem VIII.1 and provide the parts of the proof of Theorem VIII.1 which have not occurred in the literature.

We let p denote an arbitrary prime for the moment, abbreviate $F = K(x)$ and let $f \in F$ be a rational function. A simple Artin–Schreier extension denoted by E_f , is given by adjoining to F a root of the polynomial $y^p - y - f \in F[y]$. Examples of such extensions are the function fields of elliptic curves in characteristics two and three.

The Artin–Schreier operator is denoted by $\wp(y) = y^p - y$. We then also write $F(\wp^{-1}(f))$ for E_f and $\wp(F) = \{f^p - f : f \in F\}$. More generally, Theorem VIII.1 uses the following construction and theorem, which is a special version of [260, p. 279, Theorem 3.3]:

THEOREM VIII.12. *Let $\bar{F}$ be a fixed separable closure of F . For every additive subgroup $\Delta \leq F$ with $\wp(F) \subseteq \Delta \subseteq F$ there is a field $C = F(\wp^{-1}(\Delta))$ with $F \subseteq C \subseteq \bar{F}$ obtained by adjoining all roots of all polynomials $y^p - y - d$ for $d \in \Delta$ in $\bar{F}$ to F . Given this, the map*

$$\Delta \mapsto C = F(\wp^{-1}(\Delta))$$

defines a one-to-one correspondence between such additive subgroups Δ and abelian extensions C/F in $\bar{F}$ of exponent p .

For our purposes this construction is only applied for very special Δ , which will be introduced in a moment. As in Section VIII.1.2, by a Frobenius automorphism with respect to K/k of a function field over K we mean an automorphism of order $n = [K : k]$ of that function field which extends the Frobenius automorphism of K/k . Raising the coefficients of a rational function in $F = K(x)$ to the q th power yields for example a Frobenius automorphism of F with respect to K/k , which we denote by σ .

For $f \in F$ we define $\Delta_f := \{d^p - d + \sum_{i=0}^{n-1} \lambda_i \sigma^i(f) : d \in F \text{ and } \lambda_i \in \mathbb{F}_p\}$. This is the subgroup of the additive group of F which is generated by f and contains $\wp(F)$. Also, let $m_f = \sum_{i=0}^m \lambda_i t^i$ with $\lambda_m = 1$ be the unique

polynomial of smallest degree in $\mathbb{F}_p[t]$ such that $\sum_{i=0}^m \lambda_i \sigma^i(f) = d^p - d$ for some $d \in F$. As in Section VIII.2.4, we can define a multiplication of polynomials $h(t) = \sum_{i=0}^d \lambda_i t^i \in k[t]$ and rational functions $z \in F$ by $h(t)z = \sum_{i=0}^d \lambda_i \sigma^i(z)$. This makes the additive group of F into a $k[t]$ -module. The polynomials $m_f(t)$ are then by definition polynomials in $\mathbb{F}_p[t]$ of smallest degree such that $m_f(t)f \in \wp(F)$. The field $F(\wp^{-1}(\Delta_f))$ exists by Theorem VIII.12 and has degree $p^{\deg(m_f)}$ over F . Further statements on the existence of a Frobenius automorphism of this field with respect to K/k and its genus can be found in [168].

We now let $p = 2$ and $f = \gamma/x + \alpha + \beta x$. The field $F(\wp^{-1}(\Delta_f))$ is then equal to the field C of Theorem VIII.1 defined in Section VIII.2.1. Let us exhibit an explicit model for C . We have

$$m_f = \begin{cases} \text{lcm}\{m_\gamma, m_\beta\} & \text{if } \text{Tr}_{K/\mathbb{F}_2}(\alpha) = 0, \\ \text{lcm}\{m_\gamma, m_\beta, t + 1\} & \text{otherwise.} \end{cases} \quad (\text{VIII.5})$$

Let $m = \deg(m_f)$. The classes of $\sigma^i(f)$ for $0 \leq i \leq m - 1$ form an $\mathbb{F}_2$ -basis of $\Delta_f/\wp(F)$. From Theorem VIII.12 it follows that C is obtained by adjoining one root of every $y^2 - y - \sigma^i(f)$ to F . In other words, $C = F[y_0, \dots, y_{m-1}]/I$, where I is the ideal of the polynomial ring $F[y_0, \dots, y_{m-1}]$ generated by the polynomials $y_i^2 - y_i - \sigma^i(f)$ for $0 \leq i \leq m - 1$. We write $\bar{y}_i$ for the images of the y_i in C and abbreviate $\bar{y} = \bar{y}_0$.

Using this notation we are able to prove Theorem VIII.1(iii) and (iv). Assume without loss of generality that $\deg(m_\gamma) \leq \deg(m_\beta)$. Let $h \in \mathbb{F}_2[t]$. The field C then contains the roots of $y^2 - y - h(t)f$ and is generated by these roots for all h of degree less than m . Using polynomial division write $h = sm_\gamma + r$ with $s, r \in \mathbb{F}_2[t]$ and $\deg(r) < \deg(m_\gamma)$. Thus $h(t)f = s(t)(m_\gamma(t)f) + r(t)f$. The rational function $m_\gamma(t)f$ is of the form $\rho + \delta x$ with $\rho, \delta \in K$. As a result, $h(t)f$ is an $\mathbb{F}_2$ -linear combination of the conjugates $\sigma^i(f)$ for $0 \leq i \leq \deg(m_\gamma) - 1$ and $\sigma^j(m_\gamma(t)f)$ for $0 \leq j \leq m - \deg(m_\gamma) - 1$. We write $\bar{w}_j$ for a root of the polynomials $w_j^2 - w_j - \sigma^j(m_\gamma(t)f)$ in C . Then $C = F[\bar{y}_0, \dots, \bar{y}_{\deg(m_\gamma)-1}, \bar{w}_0, \dots, \bar{w}_{m-\deg(m_\gamma)-1}]$. By [145, Lemma 7] the field $L = F[\bar{w}_0, \dots, \bar{w}_{m-\deg(m_\gamma)-1}]$ is a rational function field and the extension L/F has degree $2^{m-\deg(m_\gamma)}$. Since C/F has degree 2^m , we see that L has index $2^{\deg(m_\gamma)}$ in C . Furthermore, L is equal to $F(\wp^{-1}(\Delta_{m_\gamma(t)f}))$. Since $\sigma(\Delta_{m_\gamma(t)f}) = \Delta_{m_\gamma(t)f}$ we have that the Frobenius automorphism σ on C restricts to a Frobenius automorphism on L . It thus follows that the fixed field L^0 of σ in L is a rational function field over k and has index $2^{\deg(m_\gamma)}$ in C^0 . This proves Theorem VIII.1(iii).

The hyperellipticity in Theorem VIII.1(iv) is proven in [145] and can be seen as follows. If $\gamma \in k$, then $m_\gamma = t - 1$. Consequently, by (iii) we see that C^0 contains a rational subfield of index 2, which yields the statement.

VIII.3. Extending the GHS Attack Using Isogenies

VIII.3.1. The Basic Idea. The GHS attack can be extended to a much larger class of curves by using isogenies. Consider two elliptic curves $\mathcal{E}$ and $\mathcal{E}'$ defined over K . An isogeny $\phi : \mathcal{E} \rightarrow \mathcal{E}'$ defined over K homomorphically maps points in $\mathcal{E}(K)$ to points in $\mathcal{E}'(K)$. The coordinates of an image point are defined by algebraic expressions involving the coordinates of the input point and elements of K , in a similar way to the addition formulae. In particular, a discrete logarithm problem $P = \lambda Q$ is mapped to the discrete logarithm problem $\phi(P) = \lambda\phi(Q)$. The basic idea is that $\mathcal{E}'$ might be susceptible to the GHS attack while $\mathcal{E}$ is not. In this case we would transfer the discrete logarithm problem on $\mathcal{E}$ to $\mathcal{E}'$ using ϕ and attempt to solve it there.

For every isogeny $\phi : \mathcal{E} \rightarrow \mathcal{E}'$ defined over K there is a dual isogeny $\hat{\phi} : \mathcal{E}' \rightarrow \mathcal{E}$ defined over K and $\mathcal{E}$ is said to be isogenous to $\mathcal{E}'$. This yields an equivalence relation so that elliptic curves defined over K can be partitioned into isogeny classes. Also, the isogeny class of an elliptic curve defined over K is uniquely determined by $\#\mathcal{E}(K) = q^n + 1 - t$ and its cardinality is roughly, and on average, of the order $(4q^n - t^2)^{1/2}$, where t satisfies $|t| \leq 2q^{n/2}$ (more precisely; the cardinality is equal to the Kronecker–Hurwitz class number $H(t^2 - 4q^n)$, see [136], [293]).

If an isogeny class contains an elliptic curve which is susceptible to the GHS attack, the whole isogeny class can be considered susceptible using the above idea. Of course this raises the following two questions: Can it be efficiently determined whether an isogeny class contains a susceptible curve and if so, can the isogeny be efficiently computed? There are two main strategies:

Isogeny Strategy 1. For all $\mathcal{E}'$ which are susceptible to the GHS attack check whether $\#\mathcal{E}'(K) = \#\mathcal{E}(K)$. If so, compute the isogeny between $\mathcal{E}$ and $\mathcal{E}'$.

Isogeny Strategy 2. Compute random (all) $\mathcal{E}'$ in the isogeny class of $\mathcal{E}$ and the corresponding isogenies from $\mathcal{E}$ to $\mathcal{E}'$. Check whether one of the $\mathcal{E}'$ is susceptible to the GHS attack.

Again, we only consider elliptic curves in the unique form (VIII.1) so that we are effectively dealing here and in the following with isomorphism classes of elliptic curves.

Assuming that the cardinality of the isogeny class of $\mathcal{E}$ is roughly $q^{n/2}$ and that isogeny class membership and being susceptible to the GHS attack are basically independent properties, we expect that Strategy 1 is more efficient in terms of checks than Strategy 2 if the number of the susceptible $\mathcal{E}'$ is less than $q^{n/2}$, and that Strategy 2 is more efficient otherwise.

VIII.3.2. Isogeny Probabilities. Let $S'_{m_1(t), m_2(t)}$ denote a system of conjugacy class representatives of the operation of the 2-power Frobenius on

$S_{m_1(t), m_2(t)}$. In order to obtain some quantitative statements about the above strategies we assume that $S'_{m_1(t), m_2(t)}$ behaves like any randomly and uniformly chosen set of ordinary elliptic curves defined over K of the same cardinality. We want to estimate the probability that a randomly and uniformly chosen elliptic curve $\mathcal{E}$ is isogenous to a fixed subset of N elliptic curves $\mathcal{E}_i$ from $S'_{m_1(t), m_2(t)}$.

LEMMA VIII.13. *Let $\mathcal{E}, \mathcal{E}_1, \dots, \mathcal{E}_N$ be randomly, uniformly and independently chosen ordinary elliptic curves. Then*

$$\Pr(\mathcal{E} \sim \mathcal{E}_1 \text{ or } \dots \text{ or } \mathcal{E} \sim \mathcal{E}_N) \geq 1 - \left(1 - \frac{1}{(1 - 1/p)4q^{n/2} + 2}\right)^N.$$

PROOF. Abbreviate $A_N = \Pr(\mathcal{E} \sim \mathcal{E}_1 \text{ or } \dots \text{ or } \mathcal{E} \sim \mathcal{E}_N)$ and let $\mathcal{E}'$ denote a further randomly, uniformly and independently chosen ordinary elliptic curve. Using the complementary event it is straightforward to prove that

$$A_N = 1 - (1 - \Pr(\mathcal{E} \sim \mathcal{E}'))^N. \quad (\text{VIII.6})$$

If a_i are M non-negative numbers such that $\sum_{i=1}^M a_i = 1$, then $\sum_{i=1}^M a_i^2 \geq 1/M$. This is easily seen by writing $a_i = 1/M + \varepsilon_i$ with $\sum_i \varepsilon_i = 0$ and expanding this in $\sum_{i=1}^M a_i^2$.

The following sums are over all ordinary isogeny classes I and the symbol $\#\{I\}$ denotes the number of such classes. Using the above observation with $a_i = \Pr(\mathcal{E} \in I)$ and $M = \#\{I\}$ we see

$$\begin{aligned} \Pr(\mathcal{E} \sim \mathcal{E}') &= \sum_I \Pr(\mathcal{E} \in I \text{ and } \mathcal{E}' \in I) = \sum_I \Pr(\mathcal{E} \in I) \Pr(\mathcal{E}' \in I) \\ &= \sum_I \Pr(\mathcal{E} \in I)^2 \geq 1/\#\{I\}. \end{aligned} \quad (\text{VIII.7})$$

The number of ordinary isogeny classes satisfies

$$\#\{I\} \leq (1 - 1/p)4q^{n/2} + 2. \quad (\text{VIII.8})$$

This is because every integer $q^n + 1 - t$ with $t^2 \leq 4q^n$ and $p \nmid t$ occurs as the number of points of an ordinary elliptic curve over K (see [344]).

Combining (VIII.8), (VIII.7) and (VIII.6) yields the lemma. $\square$

According to Lemma VIII.13, it is reasonable to expect that a randomly and uniformly chosen elliptic curve $\mathcal{E}$ will be isogenous to at least one of the $\mathcal{E}_i$ from S'_{m_1, m_2} with probability approximately at least $N/(2q^{n/2})$, if N is much less than $q^{n/2}$, and with probability approximately one, if N is much larger than $q^{n/2}$. The first probability can in fact be improved slightly when one restricts for example to elliptic curves (VIII.1) with $a = 0$. Recall that if $\text{Tr}_{K/\mathbb{F}_2}(a) = 0$, then the group order $\#E(K)$ of the elliptic curve is congruent to 0 modulo 4 and if $\text{Tr}_{K/\mathbb{F}_2}(a) = 1$, then it is congruent to 2 modulo 4 (see [ECC, p. 38]). Thus elliptic curves with $a = 0$ represent only about half of all isogeny classes. It follows that the first of the above probabilities

improves by a factor of two. Another property in this line is $\#E(K) = 0 \bmod 8$ if and only if $\text{Tr}_{K/\mathbb{F}_2}(b) = 0$ for $a = 0$ and $nr \geq 3$ ([242]).

By our assumption on S'_{m_1, m_2} and in view of Isogeny Strategy 2 we expect that membership in S'_{m_1, m_2} and being isogenous to a randomly and uniformly chosen elliptic curve $\mathcal{E}$ are approximately independent events if $\#S'_{m_1, m_2}$ is much larger than $q^{n/2}$. More precisely, we expect $\#(S'_{m_1, m_2} \cap I) \approx \#S'_{m_1, m_2} \#I / (2q^n)$ since the curves from S'_{m_1, m_2} should distribute over the isogeny classes relative to their sizes.

VIII.3.3. Computing Isogenies. Isogeny Strategies 1 and 2 require the computation of isogenies. In Isogeny Strategy 1 we have to construct the isogeny between $\mathcal{E}$ and $\mathcal{E}'$ given only that $\#\mathcal{E}(K) = \#\mathcal{E}'(K)$. In Isogeny Strategy 2 we start with $\mathcal{E}$ and have to construct the isogeny to a randomly and uniformly chosen $\mathcal{E}'$ of the unique form (VIII.1) in the isogeny class of $\mathcal{E}$.

We recall some additional basic facts about isogenies and endomorphism rings for *ordinary* elliptic curves. Useful references for the required theory and algorithmic aspects are [238, 307, 293, 129, 134, 207].

Endomorphisms of $\mathcal{E}$ are isogenies of $\mathcal{E}$ to itself. Addition and composition of maps make the set of endomorphisms into a ring $\text{End}(\mathcal{E})$. For elliptic curves defined over K all endomorphisms are defined over K as well. The q^n th power Frobenius endomorphism $\pi : (x, y) \mapsto (x^{q^n}, y^{q^n})$ satisfies $\pi^2 - t\pi + q^n = 0$ with $t = q^n + 1 - \#\mathcal{E}(K)$, $t^2 \leq 4q^n$ and $t \not\equiv 0 \bmod p$, so $\mathbb{Q}(\pi)$ is an imaginary quadratic number field. The ring $\text{End}(\mathcal{E})$ is an order in $\mathbb{Q}(\pi)$ with $\mathbb{Z}[\pi] \subseteq \text{End}(\mathcal{E}) \subseteq \mathcal{O}_{\max}$, where $\mathcal{O}_{\max}$ is the maximal order or ring of algebraic integers of $\mathbb{Q}(\pi)$. Conversely, if $\pi^2 - t\pi + q^n = 0$, $t^2 \leq 4q^n$ and $t \not\equiv 0 \bmod p$ for some algebraic integer π , then every order $\mathcal{O}$ with $\mathbb{Z}[\pi] \subseteq \mathcal{O} \subseteq \mathcal{O}_{\max}$ occurs as the endomorphism ring of an elliptic curve $\mathcal{E}$ defined over K with $\#\mathcal{E}(K) = q + 1 - t$, such that π corresponds to the q^n th power Frobenius endomorphism. There is a bijection of positive integers d and suborders $\mathcal{O}_d$ of $\mathcal{O}_{\max}$ of index d , given by $d \mapsto \mathcal{O}_d = \mathbb{Z} + d\mathcal{O}_{\max}$. The integer d is called conductor of $\mathcal{O}_d$.

The degree of an isogeny $\phi : \mathcal{E} \rightarrow \mathcal{E}'$ defined over K is equal to the degree of the function field extension $K(\mathcal{E})/K(\mathcal{E}')$ defined by ϕ and coincides roughly with the degrees of its defining algebraic expressions. The degree of isogenies is multiplicative with respect to composition. Every isogeny defined over K can be decomposed into a chain of isogenies of prime degree defined over K . The cardinality of the kernel of ϕ as a homomorphism from $\mathcal{E}(K)$ to $\mathcal{E}'(K)$ is bounded by the degree of ϕ (and equal to the degree if ϕ is separable). An isomorphism is an isogeny of degree one. For every $\mathcal{E}$ defined over K there is an $\mathcal{E}'$ defined over K (a quadratic twist of $\mathcal{E}$) such that for the j -invariants $j(\mathcal{E}) = j(\mathcal{E}')$, $\mathcal{E}$ and $\mathcal{E}'$ are isomorphic over a quadratic extension of K , and every $\mathcal{E}_2$ defined over K with $j(\mathcal{E}_2) = j(\mathcal{E})$ is isomorphic over K to $\mathcal{E}$ or $\mathcal{E}'$. Furthermore, $\#\mathcal{E}(K) + \#\mathcal{E}'(K) = 2q^n + 2$. If ϕ is of degree r , then $\phi_r(j(\mathcal{E}), j(\mathcal{E}')) = 0$ where ϕ_r is the r th modular polynomial. Conversely, if

$\phi_r(j(\mathcal{E}), x) = 0$ for $x \in K$, then there is an $\mathcal{E}'$ defined over K with $j(\mathcal{E}') = x$ and an isogeny $\phi : \mathcal{E} \rightarrow \mathcal{E}'$ defined over K of degree r .

If $\mathcal{E}$ and $\mathcal{E}'$ are isogenous, then $\text{End}(\mathcal{E})$ and $\text{End}(\mathcal{E}')$ are (isomorphic to) orders within the same imaginary quadratic number field. More specifically it holds that $\text{End}(\mathcal{E}) = \text{End}(\mathcal{E}')$, $\text{End}(\mathcal{E}) \subseteq \text{End}(\mathcal{E}')$ with $(\text{End}(\mathcal{E}') : \text{End}(\mathcal{E})) = \ell$ or $\text{End}(\mathcal{E}') \subseteq \text{End}(\mathcal{E})$ with $(\text{End}(\mathcal{E}) : \text{End}(\mathcal{E}')) = \ell$, if there is an isogeny $\phi : \mathcal{E} \rightarrow \mathcal{E}'$ of prime degree ℓ . The cases where the index changes can be (partly) read off the number of zeros of $\phi_\ell(j(\mathcal{E}), x)$. An isogeny $\phi : \mathcal{E} \rightarrow \mathcal{E}'$ of elliptic curves with $\mathcal{O} = \text{End}(\mathcal{E}) = \text{End}(\mathcal{E}')$ defines a uniquely determined invertible (and integral) ideal I of $\mathcal{O}$ with $N(I) = \deg(\phi)$, and any such I is obtained this way. Isomorphisms ϕ correspond to $I = \mathcal{O}$. Composition of isogenies corresponds to ideal multiplication. If $\phi_1 : \mathcal{E} \rightarrow \mathcal{E}_1$ and $\phi_2 : \mathcal{E} \rightarrow \mathcal{E}_2$ are isogenies with the ideals I_1 and I_2 , then $\mathcal{E}_1$ is isomorphic to $\mathcal{E}_2$ if and only if I_1/I_2 is principal. As a result, the isogeny structure between the isomorphism classes is equivalent to the group structure of $\text{Pic}(\mathcal{O})$, the group of classes of invertible ideals of $\mathcal{O}$ modulo principal ideals.

The basic technique in Isogeny Strategies 1 and 2 is as follows. For $\mathcal{O} = \text{End}(\mathcal{E})$ we can use the group structure of $\text{Pic}(\mathcal{O})$ for random walks along chains of isogenies of prime degree starting at $\mathcal{E}$. This way we can generate random elliptic curves $\mathcal{E}'$ with $\text{End}(\mathcal{E}') = \mathcal{O}$ and known isogenies $\mathcal{E} \rightarrow \mathcal{E}'$ for Isogeny Strategy 2. For Isogeny Strategy 1 we use a second random walk starting at the second elliptic curve $\mathcal{E}'$, assuming $\text{End}(\mathcal{E}') = \mathcal{O}$ for the moment. If these walks meet, we can connect the two parts using dual isogenies to obtain a chain of isogenies from $\mathcal{E}$ to $\mathcal{E}'$. Since we consider only elliptic curves of the unique form (VIII.1) we are effectively working with isomorphism classes.

A single step in the random walk from a curve $\mathcal{E}_i$ to a curve $\mathcal{E}_{i+1}$ to be determined proceeds as follows. We assume that $\mathcal{O} = \text{End}(\mathcal{E})$ is known and $\text{End}(\mathcal{E}_i) = \mathcal{O}$ holds, where $\mathcal{O}$ can be computed by the algorithm of Kohel [207]. A prime number ℓ is chosen which does not divide the conductor of $\mathcal{O} = \text{End}(\mathcal{E}_i)$ and is split or ramified in $\mathcal{O}$. The j -invariants of the curves $\mathcal{E}_{i+1}$ related to $\mathcal{E}_i$ by an isogeny of degree ℓ are the roots x of $\phi_\ell(j(\mathcal{E}_i), x)$ in K . If (and only if) ℓ divides the index of $\mathbb{Z}[\pi]$ in $\mathcal{O}$, then not all of these roots result in an $\mathcal{E}_{i+1}$ with $\text{End}(\mathcal{E}_{i+1}) = \text{End}(\mathcal{E}_i)$; the case $(\text{End}(\mathcal{E}_i) : \text{End}(\mathcal{E}_{i+1})) = \ell$ is also possible. Using the techniques of [207, p. 46] we determine those x for which $\mathcal{E}_{i+1}$ isogenous to $\mathcal{E}_i$ has the same endomorphism ring. According to whether ℓ is split or ramified, there are one or two such values. We choose a value and compute $\mathcal{E}_{i+1}$ of the unique form (VIII.1) and the isogeny from it. Checking the action of the Frobenius or another suitable endomorphism on the kernel of the isogeny allows one to determine the prime ideal of norm ℓ above ℓ which corresponds to the isogeny [ECC, 294].

For the whole random walk we note that any class in $\text{Pic}(\mathcal{O})$ can be represented by an ideal of $\mathcal{O}$ of (small) norm $O(|d(\mathcal{O})|^{1/2})$, where $d(\mathcal{O})$ denotes the discriminant of $\mathcal{O}$ and is $O(q^n)$. Furthermore, $\#\text{Pic}(\mathcal{O}) \approx O(|d(\mathcal{O})|^{1/2})$ and $\text{Pic}(\mathcal{O})$ is generated by split prime ideals of (very small) norm $O(\log(|d(\mathcal{O})|)^2)$ under the GRH (generalized Riemann hypothesis). We let B denote a set of split or ramified prime ideals of very small norm which generate $\text{Pic}(\mathcal{O})$, and let B_0 the corresponding prime numbers. The prime numbers ℓ in every single step are chosen from B_0 such that the walk extends over the whole of $\text{Pic}(\mathcal{O})$. Furthermore, the i th step of the random walk starting at $\mathcal{E}$ can be represented by a single ideal I_i of small norm, corresponding to an isogeny $\mathcal{E} \rightarrow \mathcal{E}_i$. The ideal I_i is defined inductively as follows. First, $I_0 = \mathcal{O}$. For the $(i+1)$ -th step the curve $\mathcal{E}_{i+1}$, an isogeny and a prime ideal P are computed as above. Then I_{i+1} is defined to be a representative of small norm of the class of $I_i P$ and corresponds to an isogeny $\mathcal{E} \rightarrow \mathcal{E}_{i+1}$.

Assume the random walk stops at $\mathcal{E}_r$, and the isogeny $\mathcal{E} \rightarrow \mathcal{E}_r$ is described by I_r or a chain of isogenies of prime degree. The factorization of I_r is likely to contain prime ideals which have too large a norm for our purpose. Also, the chain may have length exponential in $n \log(q)$. We can obtain a much reduced chain as follows. Using techniques from index-calculus in imaginary quadratic orders we compute a representation of I_r in $\text{Pic}(\mathcal{O})$ in terms of powers of the elements of B or an enlargement of B in the form $I_r = (\gamma) \prod_{i=1}^m P_i^{d_i}$, where m and the d_i are (expected to be) polynomial in $n \log(q)$ if B is sufficiently large. Again using techniques from point counting we determine a chain of explicitly given isogenies from $\mathcal{E}$ to $\mathcal{E}_r$, such that every step corresponds to a P_i . This concludes the description of the random walks on (isomorphism classes of) elliptic curves with endomorphism ring $\mathcal{O}$.

In Isogeny Strategy 2 we do not want to restrict the possible endomorphism rings to a given $\mathcal{O}$. In a precomputation we compute all intermediate orders $\mathcal{O}_\nu$ with $\mathbb{Z}[\pi] \subseteq \mathcal{O}_\nu \subseteq \mathcal{O}_{\max}$ and isogenies $\mathcal{E} \rightarrow \mathcal{E}_\nu$ to curves $\mathcal{E}_\nu$ with $\mathcal{O}_\nu = \text{End}(\mathcal{E}_\nu)$, using [207]. We start random walks in every $\mathcal{E}_\nu$. In every step we choose ν with probability about $\sum_{\mu \neq \nu} \#\text{Pic}(\mathcal{O}_\mu) / \sum_{\mu} \#\text{Pic}(\mathcal{O}_\mu)$ and extend the walk from say $\mathcal{E}_{\nu,i}$ to some $\mathcal{E}_{\nu,i+1}$. The curve $\mathcal{E}_{\nu,i+1}$ is returned.

In Isogeny Strategy 1 the curves $\mathcal{E}$ and $\mathcal{E}'$ may have different endomorphism rings. In a precomputation we compute isogenies $\mathcal{E} \rightarrow \mathcal{E}_0$ and $\mathcal{E}' \rightarrow \mathcal{E}'_0$ such that $\text{End}(\mathcal{E}_0)$ and $\text{End}(\mathcal{E}'_0)$ are equal to $\mathcal{O} = \mathcal{O}_{\max}$. Following the Pollard methods we start a random walk at $\mathcal{E}_0$ of length $t = O(\#\text{Pic}(\mathcal{O})^{1/2}) = O(q^{n/4})$. Here $\ell, x = j(\mathcal{E}_{i+1})$ and hence the prime ideal P are chosen in a way that depends deterministically on $j(\mathcal{E}_i)$ and gives a pseudorandom distribution close to uniform. Then we proceed analogously with a random walk starting at $\mathcal{E}'_0$. After an expected t steps we find s such that $j(\mathcal{E}_t) = j(\mathcal{E}'_s)$. Since $\#\mathcal{E}(K) = \#\mathcal{E}'(K)$, the curves $\mathcal{E}_t$ and $\mathcal{E}'_s$ must be isomorphic, so $I = I_t/I'_s$ corresponds to an isogeny between $\mathcal{E}_0$ and $\mathcal{E}'_0$, where I_t and I'_s are the ideals describing the random walks as above. Applying the index-calculus trick for

the reduction of random walks to I and combining this with the isogenies from the precomputation (and their duals) finally yields a short chain of isogenies $\mathcal{E} \rightarrow \mathcal{E}'$.

In many cases $\mathbb{Z}[\pi] = \mathcal{O}_{\max}$, so that the complications with intermediate orders in Isogeny Strategy 1 and Isogeny Strategy 2 do not occur.

THEOREM VIII.14. *Let $\mathcal{E}$ and $\mathcal{E}'$ be two ordinary isogenous elliptic curves such that $\#\mathcal{E}(K) = \#\mathcal{E}'(K) = q^n + 1 - t$, and let l be the largest prime or one with l^2 dividing $(4q^n - t^2)$. Under the GRH and further reasonable assumptions there is a probabilistic algorithm which computes $O(n \log(q))$ isogenies $\phi_i : \mathcal{E}_i \rightarrow \mathcal{E}_{i+1}$ of degree $O(\max\{(n \log(q))^2, l\})$ such that $\phi = \prod_i \phi_i$ is an isogeny between $\mathcal{E}$ and $\mathcal{E}'$. The expected running time is $O(\max\{q^{n/4+\varepsilon}, l^{3+\varepsilon}\})$.*

The theorem follows from [129], [134] along the lines explained above. The algorithm involves some not rigorously proven steps from index-calculus in imaginary quadratic orders which accounts for the GRH and further “reasonable” assumptions. In most cases l will be fairly small, so that the running time of the algorithm is essentially $O(q^{n/4+\varepsilon})$. A worse running time can only occur when l is large since potentially some isogenies ϕ_i of degree l could be required. If $\text{End}(\mathcal{E})$ and $\text{End}(\mathcal{E}')$ are equal, then using isogenies of degree l can be circumvented; but if the mutual index contains a large prime l , isogenies of degree l cannot be avoided. The algorithm is particularly efficient if $4q^n - t^2$ is small or if $\mathcal{O}_{\max}$ has a small class number $\#\text{Pic}(\mathcal{O}_{\max})$ and smooth index ($\mathcal{O}_{\max} : \mathbb{Z}[\pi]$).

If $\mathcal{E}$ is our target curve and $\mathcal{E}' \in S_{m_1(t), m_2(t)}$ is isogenous to $\mathcal{E}$ we can hence compute the isogeny ϕ between $\mathcal{E}$ and $\mathcal{E}'$ in (much) less time than the Pollard methods require for solving the DLP on $\mathcal{E}$, assuming that $4q^n - t^2$ is only divisible by squares of primes $l = O(q^{n/6-\varepsilon})$ or that $\text{End}(\mathcal{E}) = \text{End}(\mathcal{E}')$. Then ϕ is given in the product form $\phi = \prod_i \phi_i$ and images $\phi(P)$ are computed in time about $O(\max\{(n \log(q))^7, (n \log(q))l^3\})$. Furthermore, also due to the degree bounds for the ϕ_i , the order of the kernel of ϕ cannot be divisible by the large prime factor of $\#\mathcal{E}(K)$ and hence the DLP is preserved under ϕ .

VIII.3.4. Implications for n Odd Prime. We now combine the previous observations with the results of Sections VIII.2.3–VIII.2.5 and Table VIII.1 for n an odd prime. Since the 2-power Frobenius has order nr on K , the cardinalities of the representative sets S'_{h_i, h_i} , S'_{t-1, h_i} and $S'_{t-1, (t-1)h_i}$ are at least $1/(nr)$ times the cardinalities of the sets S_{h_i, h_i} , S_{t-1, h_i} and $S_{t-1, (t-1)h_i}$. If we take N pairwise distinct elliptic curves $\mathcal{E}_i$ from these representative sets with $N \ll\ll q^{n/2}$, we expect by Lemma VIII.13 and the discussion thereafter that a randomly and uniformly chosen elliptic curve $\mathcal{E}$ will be isogenous to one of the $\mathcal{E}_i$ with probability at least $\min\{1, N/(2q^{n/2})\}$ or $\min\{1, N/q^{n/2}\}$ if the considered elliptic curves have $a = 0$.

Following Isogeny Strategy 1, we need to actually compute the $\mathcal{E}_i$. Some details on how this can be achieved are given in the appendix of [134]. For

each curve $\mathcal{E}_i$ we check $\#\mathcal{E}(K) \cdot P = \mathcal{O}$ for some random points $P \in \mathcal{E}_i(K)$. If the check fails, $\mathcal{E}_i$ is not isogenous to $\mathcal{E}$. Otherwise it is quite likely that it is and we check $\#\mathcal{E}_i(K) = \#\mathcal{E}(K)$ using fast point counting techniques. If we find $\mathcal{E}_i$ such that $\#\mathcal{E}_i(K) = \#\mathcal{E}(K)$, we are left to apply the algorithm from Theorem VIII.14. This strategy requires a time linear in N , plus a time of about $O(q^{n/4})$ for the isogeny computation.

Following Isogeny Strategy 2, we need to sample random and uniformly distributed elliptic curves $\mathcal{E}'$ from the isogeny class of $\mathcal{E}$ as described in Section VIII.3.3. We expect to compute approximately $q^n/\#S_{h_i, h_i}$, $2q^n/\#S_{t-1, h_i}$ and $2q^n/\#S_{t-1, (t-1)h_i}$ curves $\mathcal{E}'$ and isogenies $\mathcal{E} \rightarrow \mathcal{E}'$ until $\mathcal{E}'$ is isomorphic to one of the curves in S_{h_i, h_i} , S_{t-1, h_i} and $S_{t-1, (t-1)h_i}$, respectively. Table VIII.2 contains a summary.

TABLE VIII.2. Expected Probabilities that a Random $\mathcal{E}$ is Isogenous to a Curve $\mathcal{E}'$ in S_{m_γ, m_β} and Runtimes for Isogeny Strategy 1 (Excluding the $O(q^{n/4})$ Contribution) and Isogeny Strategy 2, for n Odd Prime

m_γ	m_β	$\Pr(\mathcal{E} \sim \mathcal{E}' \in S_{m_\gamma, m_\beta})$	Strat 1	Strat 2
h_i	h_i	$\min\{1, sq^{2d-1-n/2}/(2nr)\}$	$sq^{2d-1}/(2nr)$	$2q^{n-2d+1}/s$
$t-1$	h_i	$\min\{1, sq^{d-n/2}/(nr)\}$	$2sq^d/(nr)$	q^{n-d}/s
$t-1$	$(t-1)h_i$	$\min\{1, s(q-1)q^{d-n/2}/(nr)\}$	$2sq^{d+1}/(nr)$	q^{n-d-1}/s

Example 6: Consider $n = 7$. By Example 5, a proportion of about q^{-2} of all elliptic curves over $\mathbb{F}_{q^7}$ with $\alpha = 0$ leads to an efficiently computable, not necessarily hyperelliptic C^0 of genus 7. Using Isogeny Strategy 2 and the first row of Table VIII.2, we thus expect that sampling of the order of q^2 many random elliptic curves from the isogeny class of the target curve $\mathcal{E}$ yields such a C^0 .

Example 7: Consider $n = 31$. The factorization of $t^{31} - 1$ modulo 2 consists of $t-1$ and $s=6$ irreducible polynomials $h_i(t)$ of degree $d=5$, two of which are of the trinomial form of Lemma VIII.11. Using Table VIII.1 there are hence about $3q^9$, $12q^5$ and $12q^6$ elliptic curves which lead to non-hyperelliptic and hyperelliptic curves C^0 of genus 31, 31 and 32, respectively. Using Table VIII.2 the probability that a random elliptic curve lies in the isogeny class of one of these curves is $3q^{-13/2}/(31r)$, $6q^{-21/2}/(31r)$ and $6q^{-19/2}/(31r)$, respectively. Since the above cardinalities are much smaller than $q^{31/2}$ Isogeny Strategy 1 is more efficient and requires a run time of $3q^9/(31r)$, $12q^5/(31r)$

and $12q^6/(31r)$, respectively, plus $O(q^{31/4})$ for the (possible) isogeny computation.

VIII.4. Summary of Practical Implications

We now describe practical implications of the techniques of the previous sections for some values of n and fields of cryptographical sizes. We say that the GHS attack leads to a security reduction of a special family of elliptic curves or general elliptic curves if it is more efficient than the appropriate Pollard methods for these curves. As a rule of the thumb, the effectiveness of the GHS attack depends chiefly on n , q and the “specialness” of the considered elliptic curves (namely the genus of the resulting curve). With increasing n the effectiveness drops, and with increasing q or increasing “specialness” the effectiveness increases. This means for example that for sufficiently large n the set of elliptic curves for which the GHS attack is effective is in general negligibly small. Also by Theorem VIII.3 and the discussion thereafter, there is always a (significant) security reduction due to the GHS attack for $n \geq 4$ and partly for $n \geq 3$ if the field size is large enough. The general method of Theorem VIII.3 may however not readily apply to fields of cryptographical size.

Practical implications for elliptic curves in characteristic two have been investigated in [80], [134], [168], [181], [233], [241], [242], [311].

For $n = 1$ or $n = 2$ there are no elliptic curves over $\mathbb{F}_q$ and $\mathbb{F}_{q^2}$, respectively for which the GHS attack would lead to a security reduction. The case $n = 1$ is clear as $E = C^0$ and there is nothing new to consider. The case $n = 2$ yields C^0 of genus at least two. Since the Pollard methods on E are more efficient than index calculus on curves of genus at least two there is no security reduction due to the GHS attack [80].

The case $n = 3$ has not been discussed in the literature. From the results for $n = 4$ it however appears reasonable to expect no or only a minor security reduction due to the GHS attack for any elliptic curve over $\mathbb{F}_{q^3}$.

The cases $n = 4$ and $n = 5$ are discussed in detail in [311] considering low genus index-calculus methods and in [233], [242] considering high genus index-calculus methods. The conclusion for $n = 4$ is that there is (only) a minor security reduction due to the GHS attack, applicable to any elliptic curve over $\mathbb{F}_{q^4}$, and a slightly more significant security reduction for a proportion of around $1/q$ of these elliptic curves. The case $n = 5$ is particularly interesting since there is an IETF standard [RFC 2412] using the fields $\mathbb{F}_{155}$ and $\mathbb{F}_{185}$. In [311] it is concluded that an arbitrary elliptic curve over $\mathbb{F}_{155}$ is subject to only a minor security reduction. In [233] it is argued that an arbitrary elliptic curve over $\mathbb{F}_{185}$ is subject to a security reduction by a factor of 2^{16} , resulting in a security of 2^{76} instead of 2^{92} . In [242] timing estimates are given for further fields, the security reduction becomes larger as the field

size grows. For $\mathbb{F}_{q^{600}}$ the factor is for example already 2^{69} , applicable to every elliptic curve over that field.

The case $n = 6$ is partly discussed in [242], focusing on the field $\mathbb{F}_{2^{210}}$. The conclusion is that about one quarter of all elliptic curves over $\mathbb{F}_{2^{210}}$, namely those with $\text{Tr}_{\mathbb{F}_{2^{210}}}(a) = \text{Tr}_{\mathbb{F}_{2^{210}}}(b) = 0$ or equivalently $\#E(\mathbb{F}_{2^{210}}) \equiv 0 \pmod{8}$, are subject to a security reduction by a factor of 2^{20} . The attack uses isogenies and maps the DLP to hyperelliptic curves of genus 15 or 16. Alternatively, we can make use of the more general Example 4, which yields smaller genera up to 14. Note that the resulting function field C^0 is in general not hyperelliptic, so solving the DLP for C^0 will be more expensive. Precise experiments have not been carried out, but we can still expect a significant security reduction for essentially all elliptic curves over $\mathbb{F}_{q^6}$.

The case $n = 7$ has been considered in [134] using the GHS reduction with $\gamma = 1$, and it was concluded that there should be a significant security reduction for every elliptic curve over $\mathbb{F}_{q^7}$ if only C^0 could be found efficiently enough. In [242] the field $\mathbb{F}_{2^{161}}$ is briefly discussed, for which the GHS attack would yield a feasible HCDLP for genus 7 or 8 over $\mathbb{F}_{2^{23}}$. By Example 6 we expect that sampling of the order of q^2 many random elliptic curves from the isogeny class of a target elliptic curve $\mathcal{E}_{a,b}$ over $\mathbb{F}_{q^7}$ with $a = 0$ yields a not necessarily hyperelliptic C^0 over $\mathbb{F}_q$ of genus 7. Comparing against the cost of $q^{7/2}$ for the Pollard methods finding C^0 thus takes negligible time. We can hence expect a particularly significant security reduction of up to a factor of $q^{3/2}$ for all elliptic curves over $\mathbb{F}_{q^7}$. A precise analysis and whether the DLP for elliptic curves over $\mathbb{F}_{2^{161}}$ is feasible using these techniques has not been carried out yet.

The case $n = 8$ has been discussed in [233]. There is a class of approximately q^5 elliptic curves $\mathcal{E}_{a,b}$ with $\text{Tr}_{K/\mathbb{F}_2}(a) = 0$ and $m_b(t) = (t - 1)^5$ whose security is significantly reduced by the GHS attack. In [233] it is argued that using Isogeny Strategy 1 would not present a feasible method of finding such a susceptible elliptic curve isogenous to a given arbitrary target curve. Applying Isogeny Strategy 2 however would seem to require sampling approximately q^3 random curves in the isogeny class of the target curve before a susceptible curve is found. As a result the security of any elliptic curve with $\text{Tr}_{K/\mathbb{F}_2}(a) = 0$ and $n = 8$ could be reduced by a factor of approximately q . The case $n = 8$ is of particular interest since the ANSI X.962 standard [ANSI X9.62] lists in Appendix H.4 specific elliptic curves over fields of characteristic two and extension degrees 16ℓ , where $\ell \in \{11, 13, 17, 19, 23\}$. We remark that the curves are defined over $\mathbb{F}_{16}$.

The case $n = 15$ is illustrated in [233]. A striking example is $\mathbb{F}_{2^{600}}$ where the GHS attack applies to 2^{202} curves and requires about 2^{79} steps, which is much less than the 2^{299} steps for the Pollard methods.

The case $n = 31$ has been discussed in [168],[181],[233]. The existing methods do not yield a security reduction for random elliptic curves over $\mathbb{F}_{q^{31}}$,

but do yield a very significant security reduction for special curves. Some of these special curves are given as challenge curves in [233]. For example, over $\mathbb{F}_{2^{155}}$ and $\mathbb{F}_{2^{186}}$ the Pollard methods have a run time of about 2^{77} compared to 2^{37} and 2^{92} compared to 2^{42} for the GHS attack and a hyperelliptic C^0 , respectively. Example 7 shows some attack possibilities. For $\mathbb{F}_{2^{155}}$ we expect to transfer the DLP on an elliptic curve with $a = 0$ to a non-hyperelliptic curve of genus 31 with approximate probability $3q^{-13/2}/(31r) \approx 2^{-37}$ and a run time of about $3q^9/(31r) + q^{31/4} \approx 2^{40}$.

Further extension degrees and field sizes are investigated in [233]. The above discussion shows that elliptic curves over composite extension fields $\mathbb{F}_{q^n}$ with $n \in \{5, 6, 7, 8\}$ (are likely to) offer less security due to the GHS attack than expected. On the other hand, if n is a prime $\neq 127$ in the interval 100 to 600, then the GHS attack is infeasible and does not lead to a security reduction for any elliptic curve over $\mathbb{F}_{q^n}$; see [241] and Theorem VIII.5.

VIII.5. Further Topics

In this section we briefly discuss the Weil descent methodology and generalizations of the GHS attack in odd characteristic and for more general curves. We also include some further applications.

The basic ideas of Section VIII.1 and Section VIII.2.1 can be generalized to odd characteristic and to more general curves quite verbatim and are made explicit by using Kummer and Artin–Schreier constructions. Some indications for the Artin–Schreier case can already be found in Section VIII.2.6.

VIII.5.1. Kummer Constructions. The main reference here is [104], which considers the case of elliptic and hyperelliptic curves in odd characteristic with a particular emphasis on odd prime degree extension fields. Since the second roots of unity $1, -1$ are always contained in the base field, an elliptic or hyperelliptic curve $\mathcal{H} : Y^2 = f(X)$ defines a Kummer extension $H/K(X)$ of degree two where $H = K(\mathcal{H})$ is an elliptic or hyperelliptic function field. The following statements are given and proved in [104].

THEOREM VIII.15. *Let K/k be an extension of finite fields of odd characteristic and odd degree $n = \prod_p p^{n_p}$. Let H be an elliptic or hyperelliptic function field of genus g and regular over K suitable for cryptographic applications. Choose some element $x \in H$ such that $H/K(x)$ is of degree two, given by an equation of the form $y^2 = cf(x)$, where f is monic and $c \in K^\times$.*

Then, via the GHS attack, one obtains a function field C^0 regular over k or its unique quadratic extension, an extension C/H of degree 2^{m-1} for some $m \leq n$ with $C = KC^0$, and a homomorphism from $\text{Pic}_K^0(H)$ to $\text{Pic}_k^0(C^0)$ with the following properties:

- (i) *If $c = 1$, C^0/k is regular.*
- (ii) *$g_{C^0} \leq 2^{n-1}((g+1)n-2) + 1$.*

- (iii) If there exists some field L with $k \subseteq L \subseteq K$ such that $H/L(x)$ is Galois, the large subgroup of prime order is not preserved under the homomorphism.
- (iv) If there does not exist such an L , the kernel of the homomorphism contains only elements of 2-power order and

$$g_{C^0} \geq 2^{\lceil (\sum_{p, n_p \neq 0} p^{n_p}) / (2g+2) \rceil - 2} \left(\sum_{p, n_p \neq 0} p^{n_p} - 4 \right) + 1.$$

- (v) If n is prime, then additionally

$$g_{C^0} \geq 2^{\phi_2(n)-2} (n-4) + 1,$$

where $\phi_2(n)$ denotes the multiplicative order of 2 modulo n .

- (vi) If $[KC : \bar{K}(x)] \geq 2^4$, then $C/K(x)$ does not contain an intermediate field which is rational and of index 2 in C .

Let σ be the extension of the Frobenius automorphism of K/k to $K(x)$ via $\sigma(x) = x$. Let U be the (multiplicative) $\mathbb{F}_2$ -subspace of $K(x)^\times / K(x)^{\times 2}$ generated by the conjugates $\sigma^i(f)$ and let $\bar{U}$ be the $\mathbb{F}_2$ -subspace of $\bar{K}(x)^\times / \bar{K}(x)^{\times 2}$ generated by U . Then $C = KC^0$ is obtained by adjoining all square roots of class representatives of U to $K(x)$. Furthermore, $[C : K(x)] = 2^m$ and $[\bar{K}C : \bar{K}(x)] = 2^{\bar{m}}$, where $m = \dim U$ and $\bar{m} = \dim \bar{U}$. Also, $\bar{m} = m$ if and only if C/K is regular, and $\bar{m} = m - 1$ otherwise.

For $n = 5$ and $n = 7$ there are families of elliptic curves over any extension field of degree n for which g_{C^0} assumes the lower bounds 5 and 7, respectively, given by (v). There is for example an elliptic curve over $\mathbb{F}_{100000197}$ whose group order is four times a prime and which yields $g_{C^0} = 7$. Moreover, a defining polynomial for C^0 can be given in these cases. For $n = 11, 13, 17, 19, 23$ and elliptic curves we have $g_{C^0} \geq 1793, 9217, 833, 983041, 9729$. The attack is not feasible for prime $n \geq 11$.

A further study of Kummer techniques is carried out in [326] and leads to examples of attackable (or reduced security) classes of elliptic and hyperelliptic curves for $n = 2$ and $n = 3$. We summarise the examples of [104, 326, 106] in Table VIII.5.1. A nice table of smallest possible genera depending on small values of n and $g = g_H$ is given in [106]. We remark that [326] also deals with a class of superelliptic curves.

VIII.5.2. Artin–Schreier Constructions. An elliptic or hyperelliptic curve $\mathcal{H} : Y^2 + h(X)Y = f(X)$ in characteristic two defines (after a transformation similar to the one from (VIII.1) to (VIII.2)) an Artin–Schreier extension $H/K(X)$ of degree two where $H = K(\mathcal{H})$ is an elliptic or hyperelliptic function field.

A generalization of the Artin–Schreier construction for elliptic curves as in [145] to hyperelliptic curves in characteristic two was first considered in [131]. There conditions for the hyperelliptic curves are derived such that the

TABLE VIII.3. Examples of [104, 326, 106] for $a, a_i \in K \setminus k$ and $h \in k[X]$ (no Multiple Factors Allowed on the Right Hand Sides of =)

n	H	g	C^0	g_{C^0}
2	$Y^2 = (X - a)h(X)$	$\lfloor \deg(h)/2 \rfloor$	hyperell.	$2g$
3	$Y^2 = (X - a)h(X)$	$\lfloor \deg(h)/2 \rfloor$	hyperell.	$4g + 1$
3	$Y^2 = (X - a)(X - \sigma(a))h(X)$	$\lfloor (\deg(h) - 1)/2 \rfloor$	hyperell.	$4g - 1$
3	$Y^2 = \prod_{i=1}^{g+1} (X - a_i)(X - \sigma(a_i))$	g	—	$3g$
5	$Y^2 = \prod_{i \in \{0,1,2,3\}} (X - \sigma^i(a))$	1	—	5
7	$Y^2 = \prod_{i \in \{0,1,2,4\}} (X - \sigma^i(a))$	1	—	7

construction of [145] carries through in an analogous way. Some examples of the resulting curves are

$$Y^2 + XY = X^{2g+1} + \cdots + c_3X^3 + c_2X^2 + c_1X + \theta,$$

$$Y^2 + XY = X^{2g+1} + \cdots + c_3X^3 + c_2X^2 + \theta X + c_1,$$

$$Y^2 + XY = X^{2g+1} + \cdots + \theta X^3 + c_3X^2 + c_2X + c_1,$$

$$Y^2 + XY = X^{2g+1} + \cdots + \theta'X^3 + c_1X^2 + \theta X + \theta^2,$$

where $c_i \in k$, $\theta, \theta' \in K$ and θ, θ' have n distinct conjugates. A bound $g_{C^0} \leq g2^{m-1}$ for the genus of the corresponding C^0 holds, where m is defined similarly as in Section VIII.2.6.

Another family of curves is given in [327], of the form $Y^2 + h(X)Y = f(X)h(X) + (\alpha X + \beta)h(X)^2$ with $f, h \in k[X]$, $\alpha, \beta \in K$ and some further conditions. Here the genus g_{C^0} is proven to be equal to $g2^{m-1} - 1$ or $g2^{m-1}$.

A discussion of general Artin–Schreier extensions is carried out in [168]. The main consequences for elliptic curves are presented in Section VIII.1. One result for general Artin–Schreier extensions is that g_{C^0} grows exponentially in m whence the attack can only apply to very special families of curves or if n is small. A similar statement holds true for Kummer extensions.

We remark that [327] and [168] also include Artin–Schreier extensions in characteristic $p > 2$.

VIII.5.3. Kernel of Norm-Conorm Maps and Genera. We consider a generalization of the situation in Section VIII.1.2. Let E be a function field of transcendence degree one over the finite exact constant field K , C/E a finite extension and U a finite subgroup of $\text{Aut}(C)$. The fixed field of U in C is denoted by C^0 . As in Section VIII.1.2, we obtain a homomorphism of the divisor class groups $\phi : \text{Pic}_K^0(E) \rightarrow \text{Pic}_k^0(C^0)$ by $N_{C/C^0} \circ \text{Con}_{C/E}$, the conorm from E to C followed by the norm from C to C^0 . This situation is quite general; for example, we do not require U to be abelian.

THEOREM VIII.16. *The kernel of ϕ satisfies*

- (i) $\ker(\mathrm{N}_{E/E^V}) \subseteq \ker(\phi)$, where V is any subgroup of U with $VE \subseteq E$, so V restricts to a subgroup of $\mathrm{Aut}(E)$ and E^V is the fixed field of V in E .
- (ii) If the intersection $E \cap \sigma E$ is a function field and $E, \sigma E$ are linearly disjoint over $E \cap \sigma E$ for every $\sigma \in U$, then

$$[C : E] \cdot \ker \phi \subseteq \sum_{\sigma \in U \setminus \{1\}} \mathrm{Con}_{E/E \cap \sigma E}(\mathrm{Pic}_K^0(E \cap \sigma E)).$$

For example, E and σE are linearly disjoint over $E \cap \sigma E$ if at least one is Galois over $E \cap \sigma E$.

The theorem applies in particular to the Kummer and Artin–Schreier constructions discussed so far. Condition (i) basically means that for subfield curves, $\ker(\phi)$ contains the large prime factor subgroup of $\mathrm{Pic}_K^0(E)$. In condition (ii) we have that the $\mathrm{Pic}_K^0(E \cap \sigma E)$ do not contain the large prime factor subgroup since $E \cap \sigma E = K(x)$, and $[C : E]$ is also not divisible by the large prime factor. As a result, $\ker(\phi)$ does not contain the large prime factor subgroup either.

A proof of a more general version of Theorem VIII.16 is given in [168]. The case of elliptic and hyperelliptic curves has been independently dealt with in [104].

The genus of C^0 can be computed in a number of ways. A general way, which also determines the L -polynomial of C^0 , is as follows. Let G be a finite subgroup of $\mathrm{Aut}(C)$ and let H and U be subgroups of G such that H is normal in G , $H \cap U = \{1\}$ and $G = HU$. The subgroup U operates on H by conjugation. Assume further that H is elementary abelian of prime exponent l and let $\{H_\nu \mid \nu \in I\}$ be a system of representatives under the operation of U on the subgroups of H of index l for some index set I . Let U_ν be the largest subgroup of U which leaves H_ν invariant. If A is any subgroup of G , then the fixed field of A in C is denoted by C^A and the degree of the exact constant field of C^A over that of C^G by d_{C^A} .

THEOREM VIII.17. *Under the above assumptions the L -polynomials satisfy*

$$L_{C^U}(t^{d_{C^U}}) / L_{C^G}(t) = \prod_{\nu \in I} L_{C^{H_\nu U_\nu}}(t^{d_{C^{H_\nu U_\nu}}}) / L_{C^{HU_\nu}}(t^{d_{C^{HU_\nu}}}).$$

COROLLARY VIII.18. *The genera satisfy the equation*

$$d_{C^U} g_{C^U} - g_{C^G} = \sum_{\nu \in I} (d_{C^{H_\nu U_\nu}} g_{C^{H_\nu U_\nu}} - d_{C^{HU_\nu}} g_{C^{HU_\nu}}).$$

Theorem VIII.17 and Corollary VIII.18 are proved in [168]. They can be applied to the Artin–Schreier and prime degree Kummer constructions quite straightforwardly by analysing the defining groups Δ (and U as in Section VIII.5.1).

VIII.5.4. Construction of Models. The construction of explicit defining equations for C^0 obtained by the Kummer or Artin–Schreier constructions and the computation of images under ϕ by means of computer algebra systems is quite technical but does not pose principal algorithmic problems. We refer to [145, 168, 131, 104, 326, 327] for details.

VIII.5.5. Trap Door Systems. The GHS attack with isogenies from Section VIII.3 can also be used constructively for a trap door system [324]. The basic idea is as follows. A user creates a secret elliptic curve $\mathcal{E}_s$ which is susceptible to the GHS attack. The user then computes a public elliptic curve $\mathcal{E}_p$ by means of a secret, sufficiently long and random isogeny chain starting at $\mathcal{E}_s$. The curve $\mathcal{E}_s$ and the isogeny chain are submitted to a trusted authority for key escrow, while $\mathcal{E}_p$ is used as usual in elliptic curve cryptosystems. The parameters are chosen such that the Pollard methods are the most efficient way to solve the DLP on $\mathcal{E}_p$, while solving the DLP on $\mathcal{E}_s$ is much easier but still sufficiently hard. The trusted authority thus has to invest considerable computing power to decrypt which makes widespread wire tapping infeasible. For further details and parameter choices see [324].

VIII.5.6. Other Approaches. Covering techniques can also be applied when the target function field E comes from a true subfield curve. The methods described so far do not readily apply because $\sigma E = E$, in view of Theorem VIII.16. One strategy to overcome this problem is to perform a suitable change of variable such that there are n different conjugate fields $\sigma^i(E)$, so basically one considers a different σ . Accordingly, another strategy is to twist σ by an automorphism τ of order n such that there are n different conjugate fields $(\sigma\tau)^i(E)$. If E_0 is the target field defined over k , such that $E = E_0K$, then this strategy leads to the construction of suitable extensions C_0/E_0 and $\tau_0 \in \text{Aut}(C_0)$ with $\tau_0(E_0) \neq E_0$ and departs from the Kummer and Artin–Schreier paradigms. We refer to [103, 106] for details. As a consequence, with respect to an extension K/k of degree 3 and $\text{char}(k) \neq 2, 3$, the DLP (in the trace zero group) of a genus 2 curve can always be transformed into a DLP of a genus 6 curve defined over k . For a non-negligible percentage even genus 5 is possible, which leads to a more efficient attack via index calculus than by the Pollard methods.

A further approach described in [106] uses special classes of hyperelliptic curves defined over k which admit maps to elliptic curves defined over K . The genus of these hyperelliptic curves is equal to $n = [K : k]$, so these attacks would be very efficient. However, the maps between the curves are not (yet) known explicitly and upper bounds for their degrees are very large.

We close with two more remarks. In [45] the GHS construction is used to construct elliptic curves over $\mathbb{F}_2(x)$ of high rank with constant j -invariant. In [290] a covering technique is used to construct genus 2 curves defined over k with Jacobian isogenous to the Weil restriction of a large class of

elliptic curves defined over K with respect to a quadratic extension K/k of finite fields in odd characteristic. This allows for SEA point counting while avoiding patents in ECC (see also [170, 132]).

Part 4

Pairing Based Techniques

CHAPTER IX

Pairings

S. Galbraith

Pairings in elliptic curve cryptography are functions which map a pair of elliptic curve points to an element of the multiplicative group of a finite field. They have been used in several different contexts. The purpose of this chapter is to explain some of the mathematics behind pairings, to show how they may be implemented and to give some applications of them. This builds on material of III.5 and V.2 of [ECC]. Cryptographic schemes based on pairings will be described in Chapter X.

IX.1. Bilinear Pairings

Let n be a positive integer. Let G_1 and G_2 be abelian groups written in additive notation with identity element 0. Suppose that G_1 and G_2 have exponent n (i.e., $[n]P = 0$ for all $P \in G_1, G_2$). Suppose G_3 is a cyclic group of order n written in multiplicative notation with identity element 1. A *pairing* is a function

$$e : G_1 \times G_2 \longrightarrow G_3.$$

All pairings we consider will satisfy the following additional properties:

Bilinearity: For all $P, P' \in G_1$ and all $Q, Q' \in G_2$ we have

$$e(P + P', Q) = e(P, Q)e(P', Q) \text{ and } e(P, Q + Q') = e(P, Q)e(P, Q').$$

Non-degeneracy:

- For all $P \in G_1$, with $P \neq 0$, there is some $Q \in G_2$ such that $e(P, Q) \neq 1$.
- For all $Q \in G_2$, with $Q \neq 0$, there is some $P \in G_1$ such that $e(P, Q) \neq 1$.

The two examples of pairings which we will consider are the Weil and Tate pairings on elliptic curves over finite fields. First we give some easy consequences of bilinearity.

LEMMA IX.1. *Let e be a bilinear pairing as above. Let $P \in G_1$ and $Q \in G_2$. Then*

1. $e(P, 0) = e(0, Q) = 1$.
2. $e(-P, Q) = e(P, Q)^{-1} = e(P, -Q)$.
3. $e([j]P, Q) = e(P, Q)^j = e(P, [j]Q)$ for all $j \in \mathbb{Z}$.

PROOF. Property 1 follows from $e(P, Q) = e(P + 0, Q) = e(P, Q)e(0, Q)$ and similar formulae for Q . Property 2 follows from $1 = e(0, Q) = e(P + (-P), Q) = e(P, Q)e(-P, Q)$. Property 3 is then immediate. $\square$

IX.2. Divisors and Weil Reciprocity

We recall the language of divisors from algebraic geometry. General references are Silverman [307] and Washington [343]. This section may be omitted on first reading.

Let C be a curve over a perfect field K (the two important cases for our purposes are when C is an elliptic curve or is the projective line). We denote by $C(\overline{K})$ the set of all points on the curve defined over the algebraic closure of the field K . A *divisor* on C is a formal sum $D = \sum_{P \in C(\overline{K})} n_P(P)$, where $n_P \in \mathbb{Z}$ and all but finitely many n_P are zero. The divisor with all $n_P = 0$ is denoted 0. The set of divisors on C is denoted $\text{Div}_{\overline{K}}(C)$ and has a natural group structure of addition. The *support* of a divisor D is the set of all points P such that $n_P \neq 0$. The *degree* of a divisor D is $\deg(D) = \sum_P n_P$. For $\sigma \in \text{Gal}(\overline{K}/K)$ we define $D^\sigma = \sum_P n_P(\sigma(P))$. A divisor is *defined over* K if $D = D^\sigma$ for all $\sigma \in \text{Gal}(\overline{K}/K)$.

If f is a non-zero function on C , then $\text{ord}_P(f)$ counts the multiplicity of f at P , see [307, Section II.1]. Note that $\text{ord}_P(f)$ is positive when $f(P) = 0$ and is negative if f has a pole at P . The *divisor* of a non-zero function f , written (f) , is the divisor $\sum_{P \in C(\overline{K})} \text{ord}_P(f)(P)$. It follows that $(fg) = (f) + (g)$ and $(f/g) = (f) - (g)$. A *principal divisor* on C is a divisor which is equal to (f) for some function f . An important theorem of algebraic geometry states that $\deg((f)) = 0$ (see Remark II.3.7 of [307] or Theorem VII.7.9 of [226]).

A function f is *defined over* K if it can be written with all coefficients lying in the field K . If f is a non-zero function defined over K , then the divisor (f) is defined over K .

If $f \in \overline{K}^*$ is a constant, then $(f) = 0$. Conversely, if $(f) = 0$, then f must be a constant. Hence, if $(f) = (g)$, then $(g/f) = 0$ and so g is a constant multiple of f . In other words, the divisor (f) determines the function f up to a non-zero scalar multiple.

Two divisors D and D' are said to be *equivalent* (written $D' \sim D$) if $D' = D + (f)$ for some function f . If $D_1 \sim D'_1$ and $D_2 \sim D'_2$, then $(D_1 + D_2) \sim (D'_1 + D'_2)$. The *divisor class group* of a curve is the set of all divisors of degree zero up to equivalence with group structure inherited from $\text{Div}_{\overline{K}}(C)$.

The group law on an elliptic curve can be expressed in terms of divisors. Let P_1 and P_2 be two points on $E(K)$. Suppose the line between the two points P_1 and P_2 (take the tangent line if $P_1 = P_2$) has equation $l(x, y) = 0$. This line hits the elliptic curve E at some other point $S = (x_S, y_S)$. The linear polynomial $l(x, y)$ may be interpreted as a function mapping elliptic curve points to K . The function $l(x, y)$ has divisor $(l) = (P_1) + (P_2) +$

$(S) - 3(\mathcal{O})$. The vertical line $v(x) = (x - x_S)$ passes through the points S and $P_3 = P_1 + P_2$ by definition. Interpreting $v(x)$ as a function on E gives $(v) = (S) + (P_3) - 2(\mathcal{O})$. Hence the equation $P_3 = P_1 + P_2$ is the same as the divisor equality $(P_3) - (\mathcal{O}) = (P_1) - (\mathcal{O}) + (P_2) - (\mathcal{O}) - (l/v)$ and the mapping of P to the divisor class of $(P) - (\mathcal{O})$ is a group homomorphism.

The above ideas are summarized in the following result, which relates equivalence of divisors on an elliptic curve E with point addition on E .

THEOREM IX.2. *Let E be an elliptic curve over a field K . Let*

$$D = \sum_P n_P(P)$$

be a degree zero divisor on E . Then $D \sim 0$ (i.e., there is a function f such that $D = (f)$) if and only if $\sum_P [n_P]P = \mathcal{O}$ on E .

Let f be a function and let $D = \sum_P n_P(P)$ be a divisor of degree zero such that the support of D is disjoint to the support of (f) . Define

$$f(D) = \prod_P f(P)^{n_P}.$$

Note that if $g = cf$ for some constant $c \in \overline{K}^*$, then $g(D) = f(D)$ when D has degree zero. Hence, in this case the quantity $f(D)$ depends only on the divisors (f) and D . If f and D are both defined over K , then $f(D) \in K$.

The following remarkable result is used to prove various properties of the Weil and Tate pairings. It is proved in the appendix to this chapter.

THEOREM IX.3 (Weil reciprocity). *Let f and g be non-zero functions on a curve C over K . Suppose that the support of (f) and the support of (g) are disjoint. Then $f((g)) = g((f))$.*

IX.3. Definition of the Tate Pairing

We now introduce the most important pairing in elliptic curve cryptography. The Tate pairing was introduced by Tate as a rather general pairing on abelian varieties over local fields. Lichtenbaum [224] gave an interpretation in the case of Jacobians of curves over local fields which permits explicit computation. Frey and Rück [126, 125] considered the Tate pairing over finite fields and thus introduced the Tate pairing to the cryptographic community. For more background details about the Tate pairing see Frey [127].

Let E be an elliptic curve over a field K_0 . Let n be a positive integer which is coprime to the characteristic of the field K_0 . The set of n th roots of unity is defined to be $\mu_n = \{u \in \overline{K_0}^* : u^n = 1\}$. Define the field $K = K_0(\mu_n)$ to be the extension of K_0 generated by the n th roots of unity. Define

$$E(K)[n] = \{P \in E(K) : [n]P = \mathcal{O}\}$$

and

$$nE(K) = \{[n]P : P \in E(K)\}.$$

Then $E(K)[n]$ is a group of exponent n . Further, $nE(K)$ is a subgroup of $E(K)$ and the quotient group $E(K)/nE(K)$ is a group of exponent n . One may think of $E(K)/nE(K)$ as the set of equivalence classes of points in $E(K)$ under the equivalence relation $P_1 \equiv P_2$ if and only if $(P_1 - P_2) \in nE(K)$. Define

$$(K^*)^n = \{u^n : u \in K^*\}.$$

Then $(K^*)^n$ is a subgroup of K^* and the quotient group $K^*/(K^*)^n$ is a group of exponent n . The groups $K^*/(K^*)^n$ and μ_n are isomorphic.

The groups $E(K)[n]$ and $E(K)/nE(K)$ have the same number of elements, but it is not necessarily the case that the points of $E(K)[n]$ may be used as representatives for the classes in $E(K)/nE(K)$. For example, let p and r be primes such that r^4 divides $(p-1)$. There is an elliptic curve E over $\mathbb{F}_p$ with $p-1$ points which has r^4 points of order r^2 defined over $\mathbb{F}_p$. In this case $E[r] \subseteq rE(\mathbb{F}_p)$. Another example is given below.

Example: The elliptic curve $E : y^2 = x^3 + x - 3$ over $K_0 = \mathbb{F}_{11}$ has 6 points. Let $n = 3$ and write ζ_3 for a root of $t^2 + t + 1$ over $\mathbb{F}_{11}$. Then $K = K_0(\mu_n) = \mathbb{F}_{11}(\zeta_3) = \mathbb{F}_{11^2}$. One can compute that $\#E(K) = 2^2 3^3$. A basis for $E(K)[3]$ is $\{(9, 3), (10, 3 + 6\zeta_3)\}$. The point $R = (3 + 6\zeta_3, 1 + 7\zeta_3) \in E(K)$ has order 9 and satisfies $[3]R = (10, 3 + 6\zeta_3)$. Hence $(10, 3 + 6\zeta_3) \in 3E(K)$ and so $E(K)[3]$ does not represent $E(K)/3E(K)$. A basis for $E(K)/3E(K)$ is $\{(9, 3), (3 + 6\zeta_3, 1 + 7\zeta_3)\}$.

However, in many cases of relevance for cryptography one can represent $E(K)/nE(K)$ using the points of $E(K)[n]$; see Theorem IX.22.

We are now in a position to define the Tate pairing. Let $P \in E(K)[n]$ and let $Q \in E(K)$. We think of Q as representing an equivalence class in $E(K)/nE(K)$. Since $[n]P = \mathcal{O}$, it follows that there is a function f such that $(f) = n(P) - n(\mathcal{O})$. Let D be any degree zero divisor equivalent to $(Q) - (\mathcal{O})$ such that D is defined over K and the support of D is disjoint from the support of (f) . In most cases such a divisor can easily be constructed by choosing an arbitrary point $S \in E(K)$ and defining $D = (Q + S) - (S)$. Since f and D are defined over K , the value $f(D)$ is an element of K . Since the supports of (f) and D are disjoint, we have $f(D) \neq 0$, and so $f(D) \in K^*$.

The *Tate pairing* of P and Q is defined to be

$$\langle P, Q \rangle_n = f(D)$$

interpreted as an element of $K^*/(K^*)^n$.

We emphasize that values of the Tate pairing are equivalence classes. The reader is warned that we will use the symbol $=$ to denote equivalence under this relation.

Example: Consider the elliptic curve

$$E : y^2 = x(x^2 + 2x + 2)$$

over $\mathbb{F}_3$ such that $E(\mathbb{F}_3) = \{\mathcal{O}, (0, 0)\}$. Let $n = 2$, $P = (0, 0)$ and note that $\mu_2 = \{1, 2\} \subseteq \mathbb{F}_3$. Now suppose we are to compute $\langle P, P \rangle_2$.

The function x has divisor $(x) = 2(P) - 2(\mathcal{O})$. We now must find a divisor D which is equivalent to $(P) - (\mathcal{O})$ but which has support disjoint from $\{\mathcal{O}, P\}$. Since there are only two points in $E(\mathbb{F}_3)$ we cannot in this case choose $D = (P + S) - (S)$ for some $S \in E(\mathbb{F}_3)$.

Write $\mathbb{F}_{3^2} = \mathbb{F}_3(i)$, where $i^2 = -1$, and consider points on E over $\mathbb{F}_{3^2}$. The divisor $D = ((1 + i, 1 + i)) + ((1 - i, 1 - i)) - ((1, i)) - ((1, -i))$ is defined over $\mathbb{F}_3$ as a divisor. Note that $(1, i) + (1, -i) = \mathcal{O}$ in $E(\mathbb{F}_{3^2})$ and that $(1 + i, 1 + i) + (1 - i, 1 - i)$ is a point in $E(\mathbb{F}_3)$ which is not $\mathcal{O}$, and so it is P . Hence, by Theorem IX.2 we have $D \sim (P) - (\mathcal{O})$.

Finally,

$$\langle P, P \rangle_2 = x(D) = \frac{(1+i)(1-i)}{(1)(1)} = 2.$$

IX.4. Properties of the Tate Pairing

The Tate pairing is not a well defined element of K^* as it depends on the choice of D and also on the choice of representative Q of the class in $E(K)/nE(K)$. This is why we must consider values of the Tate pairing to be equivalence classes in $K^*/(K^*)^n$. The following two lemmas show that the Tate pairing is well defined as an element of $K^*/(K^*)^n$.

LEMMA IX.4. *Let f be a function such that $(f) = n(P) - n(\mathcal{O})$ and let $D' \sim D$ be degree zero divisors such that the supports of D and D' are disjoint to the support of (f) . Suppose that f , D and D' are all defined over K . Then $f(D')/f(D) \in (K^*)^n$.*

PROOF. Write $D' = D + (g)$. Then the support of (g) is disjoint to the support of (f) and

$$f(D') = f(D + (g)) = f(D)f((g)).$$

Weil reciprocity implies that

$$f((g)) = g((f)) = g(n(P) - n(\mathcal{O})) = (g(P)/g(\mathcal{O}))^n \in (K^*)^n.$$

□

LEMMA IX.5. *Let $P \in E(K)[n]$ and $Q, R \in E(K)$. Let f be a function such that $(f) = n(P) - n(\mathcal{O})$. Let D and D' be divisors defined over K with support disjoint to the support of (f) such that $D \sim (Q) - (\mathcal{O})$ and $D' \sim (Q + [n]R) - (\mathcal{O})$. Then $f(D')/f(D) \in (K^*)^n$.*

PROOF. By Theorem IX.2, we have $D' \sim D + n(R) - n(\mathcal{O})$. By Lemma IX.4 we have $f(D') = f(D + n(R) - n(\mathcal{O}))$ up to n th powers. Finally,

$$f(D + n(R) - n(\mathcal{O})) = f(D)f((R) - (\mathcal{O}))^n$$

and the result follows. $\square$

Note that the Tate pairing depends closely on the fields under consideration. If L is an extension of K , then an element $\zeta \in K^*$ may satisfy $\zeta \notin (K^*)^n$ but $\zeta \in (L^*)^n$. Similarly, a point $Q \in E(K)$ may satisfy $Q \notin nE(K)$ but $Q \in nE(L)$.

In some situations we consider a different function f to define the pairing.

LEMMA IX.6. *Up to n th powers the pairing can be defined using a function g such that $(g) = n(P + R) - n(R)$ for an arbitrary point $R \in E(K)$.*

PROOF. Let f be such that $(f) = n(P) - n(\mathcal{O})$ and suppose $R \in E(K)$. Let h be the function corresponding to the addition of P and R . In other words, $(h) = (P + R) - (R) - (P) + (\mathcal{O})$. Defining $g = fh^n$ gives $(g) = (f) + n(h) = n(P + R) - n(R)$. Let D be any divisor with support disjoint from the supports of (f) and (g) . Then $g(D) = f(D)h(D)^n$ is equal to $f(D)$ up to n th powers. $\square$

We now list the key properties of the Tate pairing (recall that $=$ and $\neq$ often refer to equivalence modulo n th powers).

THEOREM IX.7. *Let E be an elliptic curve over K_0 and let n be coprime to the characteristic of K_0 . Let $K = K_0(\mu_n)$. The Tate pairing satisfies:*

1. (Bilinearity) *For all $P, P_1, P_2 \in E(K)[n]$ and $Q, Q_1, Q_2 \in E(K)/nE(K)$,*

$$\langle P_1 + P_2, Q \rangle_n = \langle P_1, Q \rangle_n \langle P_2, Q \rangle_n$$

and

$$\langle P, Q_1 + Q_2 \rangle_n = \langle P, Q_1 \rangle_n \langle P, Q_2 \rangle_n.$$

2. (Non-degeneracy) *Suppose K is a finite field.¹ For all $P \in E(K)[n]$, $P \neq \mathcal{O}$, there is some $Q \in E(K)/nE(K)$ such that $\langle P, Q \rangle_n \neq 1$. Similarly, for all $Q \in E(K)/nE(K)$ with $Q \notin nE(K)$ there is some $P \in E(K)[n]$ such that $\langle P, Q \rangle_n \neq 1$.*
3. (Galois invariance) *If $\sigma \in \text{Gal}(\overline{K}/K_0)$, then $\langle \sigma(P), \sigma(Q) \rangle_n = \sigma(\langle P, Q \rangle_n)$.*

PROOF. We prove bilinearity with two cases:

Case 1: Let $P_1 + P_2 = P_3$ and let g be the function such that $(P_3) - (\mathcal{O}) = (P_1) - (\mathcal{O}) + (P_2) - (\mathcal{O}) + (g)$. If $(f_1) = n(P_1) - n(\mathcal{O})$ and $(f_2) = n(P_2) - n(\mathcal{O})$, then $(f_1 f_2 g^n) = n(P_3) - n(\mathcal{O})$. Let $D \sim (Q) - (\mathcal{O})$ have support disjoint to $\{P_1, P_2, P_3, \mathcal{O}\}$ (it can be shown that such a divisor always exists). Then

$$\langle P_1 + P_2, Q \rangle_n = \langle P_3, Q \rangle_n = f_1 f_2 g^n(D) = f_1(D) f_2(D) g(D)^n$$

which is equal to $\langle P_1, Q \rangle_n \langle P_2, Q \rangle_n$ in $K^*/(K^*)^n$.

¹Non-degeneracy is also proved in some other cases, for example, p -adic fields.

Case 2: Suppose $Q_1 + Q_2 = Q_3$. If $D_1 \sim (Q_1) - (\mathcal{O})$ and $D_2 \sim (Q_2) - (\mathcal{O})$, then $D_1 + D_2 \sim (Q_3) - (\mathcal{O})$ and so

$$\langle P, Q_1 + Q_2 \rangle_n = f(D_1 + D_2) = f(D_1)f(D_2) = \langle P, Q_1 \rangle_n \langle P, Q_2 \rangle_n$$

in $K^*/(K^*)^n$.

Proofs of non-degeneracy can be found in Frey-Rück [126] and Hess [167].

For the Galois-invariance, define f^σ to be the function obtained by applying σ to all the coefficients of f . Then $\sigma(f(P)) = f^\sigma(\sigma(P))$. If $(f) = n(P) - n(\mathcal{O})$, then $(f^\sigma) = n(\sigma(P)) - n(\mathcal{O})$ and $\langle \sigma(P), \sigma(Q) \rangle_n = f^\sigma(\sigma(D)) = \sigma(f(D)) = \sigma(\langle P, Q \rangle_n)$. $\square$

IX.5. The Tate Pairing over Finite Fields

Suppose that $K_0 = \mathbb{F}_q$ is a finite field. Let E be an elliptic curve defined over K_0 and let n be an integer coprime to q which divides $\#E(K_0)$. The field $K = K_0(\mu_n)$ is some finite extension $\mathbb{F}_{q^k}$. The number k is called the *embedding degree* or *security multiplier* and it is simply the smallest positive integer such that n divides $(q^k - 1)$. We stress that the number k is, strictly speaking, a function $k(q, n)$ of q and n ; however, the context is usually clear and so we will simply write k . Since k is the order of q modulo n it follows that k divides $\phi_{\text{Eul}}(n)$ (Euler phi-function). Hence, if n is prime, then k divides $(n - 1)$.

For a random field $\mathbb{F}_q$ and a random elliptic curve E , if we let n be a large divisor of $\#E(\mathbb{F}_q)$, then the embedding degree k is usually very large (almost the same number of bits as n) and so computation in the field $\mathbb{F}_{q^k}$ has exponential complexity (in terms of the input size q).

Since E is defined over $\mathbb{F}_q$, if P and Q are defined over a proper subfield of $\mathbb{F}_{q^k}$, then $\langle P, Q \rangle_n$ is also defined over the same proper subfield of $\mathbb{F}_{q^k}$. Now suppose that n is a prime, in which case our convention is to call it r . Since $\mathbb{F}_{q^k}$ is the smallest field containing both μ_r and $\mathbb{F}_q$ it follows that, for every intermediate field $\mathbb{F}_q \subseteq L \subset \mathbb{F}_{q^k}$ with $L \neq \mathbb{F}_{q^k}$, we have $L \subseteq (\mathbb{F}_{q^k}^*)^r$. These observations lead to the following trivial but important result, which tells us when the value of the Tate pairing can be trivial.

LEMMA IX.8. *Let E be an elliptic curve over $\mathbb{F}_q$ and let r be a prime. Suppose the embedding degree for E and r is $k > 1$. Let L be a proper subfield of $\mathbb{F}_{q^k}$ which contains $\mathbb{F}_q$. If $P, Q \in E(L)$, then $\langle P, Q \rangle_r \in (\mathbb{F}_{q^k}^*)^r$.*

When $K_0 = \mathbb{F}_q$ and $K = \mathbb{F}_{q^k}$, a value of the Tate pairing is an equivalence class in $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^n$ and for practical purposes we would like a unique representative of this class. The natural way to proceed is to raise this value to the power $(q^k - 1)/n$. This kills off all n th powers leaving an exact n th root of unity in $\mathbb{F}_{q^k}$. Hence for the remainder of this chapter we consider the

bilinear pairing

$$e(P, Q) = \langle P, Q \rangle_n^{(q^k-1)/n}$$

which maps into the group $\mu_n \subset \mathbb{F}_{q^k}^*$ rather than the group $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^n$.

We now give some compatibility results for the Tate pairing.

THEOREM IX.9. *Let E be an elliptic curve over $\mathbb{F}_q$. Let $n \nmid \#E(\mathbb{F}_q)$ and suppose the embedding degree corresponding to q and n is k . Let $N = hn$ be a multiple of n which divides $q^k - 1$.*

1. *Let $P \in E(\mathbb{F}_q)$ have order n and let $Q \in E(\mathbb{F}_{q^k})$. Then*

$$\langle P, Q \rangle_N^{(q^k-1)/N} = \langle P, Q \rangle_n^{(q^k-1)/n}.$$

2. *Let $P \in E(\mathbb{F}_{q^k})[N]$ and let $Q \in E(\mathbb{F}_{q^k})$. Then $\langle P, Q \rangle_N = \langle [h]P, Q \rangle_n$ up to n th powers.² Another way to say this is that*

$$\langle P, Q \rangle_N^{(q^k-1)/n} = \langle [h]P, Q \rangle_n^{(q^k-1)/n}.$$

3. *Let $\phi : E \rightarrow E'$ be an isogeny and let $\hat{\phi}$ be the dual isogeny. Then, up to n th powers,*

$$\langle \phi(P), Q \rangle_n = \langle P, \hat{\phi}(Q) \rangle_n.$$

4. *Let $\phi : E \rightarrow E'$ be an isogeny. Then, up to n th powers,*

$$\langle \phi(P), \phi(Q) \rangle_n = \langle P, Q \rangle_n^{\deg(\phi)}.$$

PROOF.

1. Write $N = hn$. Let D be a degree zero divisor equivalent to $(Q) - (\mathcal{O})$. Suppose g is a function over $\mathbb{F}_q$ such that $(g) = n(P) - n(\mathcal{O})$. Then $(g^h) = N(P) - N(\mathcal{O})$ and so

$$\langle P, Q \rangle_N^{(q^k-1)/N} = g^h(D)^{(q^k-1)/N} = g(D)^{(q^k-1)/n} = \langle P, Q \rangle_n^{(q^k-1)/n}.$$

2. By part 1

$$\langle [h]P, Q \rangle_n^{(q^k-1)/n} = \langle [h]P, Q \rangle_N^{(q^k-1)/N} = \langle P, Q \rangle_N^{h(q^k-1)/N}.$$

3. (This proof uses notation defined in the appendix to this chapter). The point $\phi(P)$ has order dividing n . Suppose $(f) = n(P) - n(\mathcal{O})$ and $D \sim (Q) - (\mathcal{O})$. Then $(\phi_*f) = \phi_*(f) = n(\phi(P)) - n(\mathcal{O})$. By Theorem III.6.1(b) of [307], $\phi^*D \sim (\hat{\phi}(Q)) - (\mathcal{O})$. Hence, by Lemma IX.26,

$$\begin{aligned} \langle \phi(P), Q \rangle_n &= (\phi_*f)(D) \\ &= f(\phi^*D) \\ &= \langle P, \hat{\phi}(Q) \rangle_n. \end{aligned}$$

4. By part 3

$$\langle \phi(P), \phi(Q) \rangle_n = \langle P, \hat{\phi}\phi(Q) \rangle_n = \langle P, [\deg(\phi)]Q \rangle_n.$$

²Since $\langle P, Q \rangle_N$ is defined up to N th powers and $\langle [h]P, Q \rangle_n$ is defined up to n th powers we can compare their values only up to n th powers.

□

IX.6. The Weil Pairing

Let E be an elliptic curve defined over K_0 and let n be an integer coprime to the characteristic of K_0 . Define $K = K_0(E[n])$ to be the field extension of K_0 generated by the coordinates of all the points in $E(\overline{K})$ of order divisible by n . The Weil pairing is a map

$$e_n : E[n] \times E[n] \rightarrow \mu_n \subseteq K^*.$$

For more information about the Weil pairing see Section III.8 of [307] or Section III.5 of [ECC]. There are two equivalent definitions of the Weil pairing (for the proof of equivalence see [63] or [173]). The definition we give is the one which is closer to the definition of the Tate pairing.

Let $P, Q \in E[n]$ and let D, D' be degree zero divisors such that the support of D and D' are disjoint and such that $D \sim (P) - (\mathcal{O})$ and $D' \sim (Q) - (\mathcal{O})$. By Theorem IX.2, there are functions f and g such that $(f) = nD$ and $(g) = nD'$. The *Weil pairing* is defined to be

$$e_n(P, Q) = f(D')/g(D).$$

The Weil pairing takes values in $\mu_n \subseteq K^*$. The following result gives some properties of the Weil pairing.

THEOREM IX.10. *The Weil pairing satisfies the following properties.*

1. (*Bilinearity*) For all $P, P', Q, Q' \in E[n]$,

$$e_n(P + P', Q) = e_n(P, Q)e_n(P', Q)$$

and

$$e_n(P, Q + Q') = e_n(P, Q)e_n(P, Q').$$

2. (*Alternating*) $e_n(P, P) = 1$ and so $e_n(P, Q) = e_n(Q, P)^{-1}$.
3. (*Non-degeneracy*) If $e_n(P, Q) = 1$ for all $Q \in E[n]$, then $P = \mathcal{O}$.
4. (*Galois invariance*) For all $\sigma \in \text{Gal}(\overline{K}/K)$

$$e_n(\sigma(P), \sigma(Q)) = \sigma(e_n(P, Q)).$$

5. (*Compatibility*) If $P \in E[nm]$ and $Q \in E[n]$, then

$$e_{nm}(P, Q) = e_n([m]P, Q).$$

6. If $\phi : E \rightarrow E'$ is an isogeny with dual $\hat{\phi}$, then

$$e_n(\phi(P), Q) = e_n(P, \hat{\phi}(Q)).$$

PROOF. All properties except non-degeneracy can be proved using Weil reciprocity and other techniques as used for the Tate pairing. For details see Propositions 8.1 and 8.2 of Silverman [307] (also see Hess [167]). □

The following corollary is easily deduced and is of fundamental importance.

COROLLARY IX.11. *Let E be an elliptic curve over a field k and let r be a prime. Let $P, Q \in E(k)[r]$ such that $P \neq \mathcal{O}$. Then Q lies in the subgroup generated by P if and only if $e_r(P, Q) = 1$.*

Note the similarity between the definition of the Weil pairing and the Tate pairing. In the Weil pairing, the term $f(D')$ is equivalent modulo n th powers to $\langle P, Q \rangle_n$ while the term $g(D)$ is equivalent modulo n th powers to $\langle Q, P \rangle_n$. Hence we can write

$$e_n(P, Q) = \frac{\langle P, Q \rangle_n}{\langle Q, P \rangle_n} \quad \text{up to } n\text{th powers.} \quad (\text{IX.1})$$

Note that this equation is only useful in the case where $\mu_n \not\subseteq (K^*)^n$.

One difference between the Weil pairing and the Tate pairing is that the Tate pairing requires working over $K_0(\mu_n)$ while the Weil pairing requires the potentially much larger field $K_0(E[n])$. The following theorem of Balasubramanian and Koblitz shows that, in the cases most relevant for cryptography, these two fields are actually the same.

THEOREM IX.12. *(Balasubramanian and Koblitz [12]) Let E be an elliptic curve over $\mathbb{F}_q$ and let r be a prime dividing $\#E(\mathbb{F}_q)$. Suppose that r does not divide $(q-1)$ and that $\gcd(r, q) = 1$. Then $E[r] \subset E(\mathbb{F}_{q^k})$ if and only if r divides $(q^k - 1)$.*

The Weil pairing can be generalized from $E[n] = \ker([n])$ to $\ker(\phi)$ where ϕ is any isogeny. In a certain case, Garefalakis [140] shows that the generalized Weil pairing is essentially equivalent to the Tate pairing.

IX.7. Non-degeneracy, Self-pairings and Distortion Maps

We now restrict to the case where $K = \mathbb{F}_{q^k}$ and $e(P, Q) = \langle P, Q \rangle_n^{(q^k-1)/n}$.

IX.7.1. Non-Degeneracy. Non-degeneracy means that, for all points $P \in E(\mathbb{F}_{q^k})[n]$, except $P = \mathcal{O}$, there is some $Q \in E(\mathbb{F}_{q^k})$ such that $e(P, Q) \neq 1$. The question we consider more closely is, for which points Q do we have $e(P, Q) \neq 1$?

For the remainder of this subsection we restrict ourselves to the case where n is a prime, and we call it r . Then $E(\mathbb{F}_{q^k})[r], E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ and $\mu_r \subset \mathbb{F}_{q^k}^*$ may all be considered as vector spaces over $\mathbb{F}_r$. Define

$$d = \dim_{\mathbb{F}_r}(E(\mathbb{F}_{q^k})[r]) = \dim_{\mathbb{F}_r}(E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})) \in \{1, 2\}.$$

Choose a basis $\{P_i\}$ for $E(\mathbb{F}_{q^k})[r]$ over $\mathbb{F}_r$ and a basis $\{Q_i\}$ for E/rE . Then the pairing can be represented as a $d \times d$ matrix M over $\mathbb{F}_r$. For any $P = \sum_i a_i P_i \in E[r]$ and any $Q = \sum_j b_j Q_j \in E(\mathbb{F}_{q^k})$ we have

$$e(P, Q) = (a_i)^t M(b_j).$$

If $\det(M) = 0$, then there is a vector v such that $vM = 0$ and this contradicts non-degeneracy. Hence $\det(M) \neq 0$.

By fixing P we may consider the pairing as an $\mathbb{F}_r$ -linear map from the quotient group $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ to μ_r . The rank-nullity theorem implies that for every such point there is a $(d-1)$ -dimensional subspace of points Q such that $e(P, Q) = 1$. Similarly, fixing Q gives an $\mathbb{F}_r$ -linear map from $E(\mathbb{F}_{q^k})[r]$ to μ_r and an analogous result follows.

We consider the two cases in turn. If $E(\mathbb{F}_{q^k})[r]$ has dimension one as an $\mathbb{F}_r$ -vector space, then so does $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ and so for all $P \in E(\mathbb{F}_{q^k})[r]$, with $P \neq \mathcal{O}$, and all $Q \in E(\mathbb{F}_{q^k})$, with $Q \notin rE(\mathbb{F}_{q^k})$ we have $e(P, Q) \neq 1$. If $E(\mathbb{F}_{q^k})[r]$ has dimension two, then for every point $P \in E(\mathbb{F}_{q^k})[r]$, $P \neq \mathcal{O}$ there is a basis $\{Q, R\}$ for $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ such that $e(P, Q) = g \neq 1$ and $e(P, R) = 1$. It follows that $e(P, [a]Q + [b]R) = g^a \neq 1$ for all a coprime to r .

IX.7.2. Self-Pairings. The alternating property of the Weil pairing is that the pairing of any point P with itself is always 1. We now consider the possibilities for a “self-pairing” in the case of the Tate pairing.

The first issue is that $E(\mathbb{F}_{q^k})[n]$ and $E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k})$ are different groups. If $P \in E(\mathbb{F}_{q^k})[n] \cap nE(\mathbb{F}_{q^k})$, then $e(P, P) = 1$. Theorem IX.22 shows that many relevant cases for cryptography have $E[n] \cap nE(\mathbb{F}_{q^k}) = \{\mathcal{O}\}$. Similarly, if $P \in E(\mathbb{F}_q)$ has prime order r and if $k > 1$, then by Lemma IX.8 $e(P, P) = 1$ and so the pairing is trivial. We have established the following fact.

LEMMA IX.13. *Let E be an elliptic curve over $\mathbb{F}_q$, let r be a prime dividing $\#E(\mathbb{F}_q)$ and let $P \in E(\mathbb{F}_q)$ be a point of order r . Let k be the embedding degree corresponding to q and r . Then $e(P, P) = 1$ if $k > 1$ or if $P \in rE(\mathbb{F}_q)$.*

In the case $k = 1$ there are several possibilities. Suppose E is an elliptic curve over $\mathbb{F}_q$ and r is a prime number such that $r \mid \#E(\mathbb{F}_q)$ and $r \mid (q-1)$ (and so $k = 1$). Let $P \neq \mathcal{O}$ be a point of order r . If $r \parallel \#E(\mathbb{F}_q)$ (if p is a prime and k is an integer, then the symbol $p^k \parallel m$ means that p^k divides m and p^{k+1} does not divide m), then there is a unique subgroup of order r in $E(\mathbb{F}_q)$ and the non-degeneracy of the Tate pairing implies that $e(P, P) \neq 1$.

If there are r^2 points of order r in $E(\mathbb{F}_q)$, then it is possible that $e(P, P) \neq 1$. For example, the curve $y^2 = x^3 + 11$ over $\mathbb{F}_{31}$ has group structure $(\mathbb{Z}/5\mathbb{Z})^2$ and the points $P = (2, 9)$ and $Q = (3, 10)$ form a basis for the points of order 5. One can compute that $e(P, P) = 16$, $e(Q, Q) = 8$, $e(P, Q) = 2$ and $e(Q, P) = 8$. It follows that

$$e([a]P + [b]Q, [a]P + [b]Q) = 2^{4a^2 + 4ab + 3b^2}$$

and hence $e(R, R) \neq 1$ for all points $R \in E[5]$ except $R = \mathcal{O}$.

An example of the opposite situation is given by $E: y^2 = x^3 + 4$ over $\mathbb{F}_{997}$ (this example was provided by Kim Nguyen). The points $P = (0, 2)$ and $Q = (747, 776)$ generate the 9 points of order 3 in $E(\mathbb{F}_{997})$. One can compute that $e(P, P) = 1$, $e(Q, Q) = 1$ and $e(P, Q) = e(Q, P)^{-1} = 304$. Hence we have $e(R, R) = 1$ for all $R \in E[3]$.

IX.7.3. Distortion Maps. The typical case in cryptographic applications is $P \in E(\mathbb{F}_q)$ and $k > 1$. It is very useful to be able to consider “self-pairings” in this case. A valuable technique introduced by Verheul [335], which applies to both the Tate and Weil pairings, is to use a non-rational endomorphism.

LEMMA IX.14. *Let $P \in E(\mathbb{F}_q)$ have prime order r and suppose $k > 1$. Suppose that $E(\mathbb{F}_{q^k})$ has no points of order r^2 . Let ϕ be an endomorphism of E . If $\phi(P) \notin E(\mathbb{F}_q)$, then $e(P, \phi(P)) \neq 1$.*

PROOF. Since ϕ is an endomorphism it follows that the order of $\phi(P)$ is r or 1. The latter case does not occur since $\phi(P) \notin E(\mathbb{F}_q)$. Hence $\{P, \phi(P)\}$ is a basis for the r -torsion on E . Since there is no r^2 -torsion it follows that $\phi(P) \notin rE(\mathbb{F}_{q^k})$. Finally, since $e(P, P) = 1$ it follows that $e(P, \phi(P)) \neq 1$. $\square$

A non-rational endomorphism which maps a point from $E(\mathbb{F}_q)$ to $E(\mathbb{F}_{q^k})$ as in the lemma is called a *distortion map*. For a given supersingular curve (see Definition IX.18) there is at least one nice choice (indeed, Theorem 5 of [336] proves that distortion maps always exist). In Section IX.13 we give examples of convenient distortion maps for supersingular curves. However, the following result from [335] implies that distortion maps do not exist for ordinary (i.e., non-supersingular; see Section IX.10) curves.

THEOREM IX.15. *Let E be an elliptic curve over $\mathbb{F}_q$ which has a distortion map. Then E is supersingular.*

PROOF. Suppose ϕ is an endomorphism which maps some point $P \in E(\mathbb{F}_q)$ to a point $\phi(P) \notin E(\mathbb{F}_q)$. Let φ be the q -power Frobenius map, which is an endomorphism of E . Then $\varphi(P) = P$ and $\varphi(\phi(P)) \neq \phi(P)$ and so $\phi \circ \varphi \neq \varphi \circ \phi$ in $\text{End}(E)$. Hence, $\text{End}(E)$ is non-commutative and E is supersingular according to Definition IX.18. $\square$

IX.7.4. The Trace Map. An alternative to using a non-rational endomorphism (which works for all elliptic curves) is the following. Let $P = (x, y) \in E(\mathbb{F}_{q^k})$ and define the *trace map*

$$\text{Tr}(P) = \sum_{\sigma \in \text{Gal}(\mathbb{F}_{q^k}/\mathbb{F}_q)} \sigma(P) = \sum_{i=0}^{k-1} (x^{q^i}, y^{q^i})$$

where the sum is elliptic curve point addition. The trace map is a group homomorphism and $\text{Tr}(P) \in E(\mathbb{F}_q)$. Under conditions like those of Lemma IX.14 it follows that if $P \in E(\mathbb{F}_{q^k})$ has prime order r , $P \notin E(\mathbb{F}_q)$ and $\text{Tr}(P) \neq \mathcal{O}$, then $e(\text{Tr}(P), P) \neq 1$. We stress that the trace map transforms points which are defined over a large field into points defined over a small field, whereas distortion maps go the other way around. For further details see Boneh, Lynn and Shacham [42].

The trace map enables mapping into a specific cyclic subgroup of $E(\mathbb{F}_{q^k})$ of order r (called the trace zero subgroup $\mathcal{T}$) as follows. If P is a randomly

chosen element of $E(\mathbb{F}_{q^k})$ of order r , then $P' = [k]P - \text{Tr}(P)$ is easily seen to satisfy $\text{Tr}(P') = \mathcal{O}$. Furthermore, if $P \notin E(\mathbb{F}_q)$ and if r is coprime to k , then $P' \neq \mathcal{O}$. This idea is used in [42]. A more efficient way to map into the trace zero subgroup using twists is given in [18]. The following degeneracy result was shown to me by Dan Boneh.

LEMMA IX.16. *Let E be an elliptic curve over $\mathbb{F}_q$. Let r be a prime such that r divides $\#E(\mathbb{F}_q)$ and such that the subgroup of r elements has embedding degree $k > 1$. Assume that r does not divide either k or $(q-1)$. Let Tr be the trace map with respect to $\mathbb{F}_{q^k}/\mathbb{F}_q$ as above and φ the q -power Frobenius. Let*

$$\mathcal{T} = \{P \in E(\mathbb{F}_{q^k})[r] : \text{Tr}(P) = \mathcal{O}\}.$$

Then $\mathcal{T} = \{P \in E(\mathbb{F}_{q^k})[r] : \varphi(P) = [q]P\}$ and for all $P, Q \in \mathcal{T}$ we have $e(P, Q) = 1$.

PROOF. If $P \in E(\mathbb{F}_q)[r]$, then $\text{Tr}(P) = [k]P$. Hence $\mathcal{T} \cap E(\mathbb{F}_q)[r] = \{\mathcal{O}\}$. On the other hand, it is easy to see that $\mathcal{T}$ is a subgroup of $E(\mathbb{F}_{q^k})[r]$. Hence $\#\mathcal{T} = r$ and $\mathcal{T}$ is cyclic.

Let φ be the q -power Frobenius map, which acts on both $\mathbb{F}_{q^k}$ and on $E(\mathbb{F}_{q^k})$. It is known that φ has eigenvalues 1 and q on $E(\mathbb{F}_{q^k})$. Let $\{P_1, P_2\}$ be a basis for $E(\mathbb{F}_{q^k})[r]$ such that $\varphi(P_1) = P_1$ and $\varphi(P_2) = [q]P_2$. Then $\text{Tr}(P_2) = [1 + q + \cdots + q^{k-1}]P_2 = [(q^k - 1)/(q - 1)]P_2 = \mathcal{O}$ and it follows that P_2 is the generator of $\mathcal{T}$.

To complete the proof it suffices to prove that $e(P_2, P_2) = 1$ and we do this by showing $e(P_2, P_2) \in \mathbb{F}_q$. Consider

$$e(P_2, P_2)^q = \varphi(e(P_2, P_2)) = e(\varphi(P_2), \varphi(P_2)) = e([q]P_2, [q]P_2) = e(P_2, P_2)^{q^2}.$$

Acting by φ^{-1} gives $e(P_2, P_2) = e(P_2, P_2)^q$, which implies $e(P_2, P_2) \in \mathbb{F}_q$. $\square$

IX.7.5. Symmetry. Another issue for the Tate pairing is to consider the relationship between $\langle P, Q \rangle_n$ and $\langle Q, P \rangle_n$, for points $P, Q \in E(K)[n]$. Up to n th powers we have

$$\langle P, Q \rangle_n = \langle Q, P \rangle_n e_n(P, Q).$$

Since the Weil pairing is non-degenerate, we often have $\langle P, Q \rangle_n \neq \langle Q, P \rangle_n$ up to n th powers.

However, if distortion or trace maps are being used, then the pairing is usually restricted to a single cyclic subgroup. In this case $Q = [m]P$ for some m and so

$$e(Q, \phi(P)) = e([m]P, \phi(P)) = e(P, [m]\phi(P)) = e(P, \phi(Q)).$$

In other words, the pairing is symmetric when restricted to a cyclic subgroup.

IX.8. Computing the Tate Pairing Using Miller's Algorithm

Victor Miller [249] gave an algorithm to compute the Weil pairing in polynomial time and this approach can also be used to compute the Tate pairing. The main issue is how to construct a function f such that $(f) = n(P) - n(\mathcal{O})$. Miller's idea is to use the double-and-add method to construct such a function in stages.

Write f_i for a function such that $(f_i) = i(P) - ([i]P) - (i-1)(\mathcal{O})$. Such a function is uniquely defined up to a constant multiple. The function we aim to compute is f_n .

LEMMA IX.17. *The functions f_i can be chosen to satisfy the following conditions.*

1. $f_1 = 1$.
2. Let l and v be the straight lines used in the computation of $[i]P + [j]P = [i+j]P$. Then

$$f_{i+j} = f_i f_j \frac{l}{v}.$$

PROOF. The fact that we can take $f_1 = 1$ is clear. In the general case we have

$$(l/v) = ([i]P) + ([j]P) - ([i+j]P) - (\mathcal{O})$$

and so

$$\begin{aligned} (f_i f_j l/v) &= i(P) - ([i]P) - (i-1)(\mathcal{O}) + j(P) - ([j]P) - (j-1)(\mathcal{O}) + (l/v) \\ &= (i+j)(P) - ([i+j]P) - (i+j-1)(\mathcal{O}). \end{aligned}$$

This proves the result. □

These formulae are most simply used in the cases $j = 1$ (addition) and $j = i$ (doubling). Miller's algorithm uses an addition chain for $[n]P$ to compute f_n . Since we are interested in the value $f_n(D)$ we evaluate all intermediate quantities at the divisor $D = (Q + S) - (S)$.³

ALGORITHM IX.1: Miller's Algorithm

INPUT: $P, Q \in E(K)$ where P has order n .

OUTPUT: $\langle P, Q \rangle_n$.

1. Choose a suitable point $S \in E(K)$.
2. $Q' \leftarrow Q + S$.
3. $T \leftarrow P$.
4. $m \leftarrow \lfloor \log_2(n) \rfloor - 1$, $f \leftarrow 1$.
5. While $m \geq 0$ do:
6. Calculate lines l and v for doubling T .
7. $T \leftarrow [2]T$.

³The generalization of Miller's algorithm to other divisors D is straightforward.

8. $f \leftarrow f^{2 \frac{l(Q')v(S)}{v(Q')l(S)}}.$
 9. If the m th bit of n is one, then:
 10. Calculate lines l and v for addition of T and P .
 11. $T \leftarrow T + P.$
 12. $f \leftarrow f^{\frac{l(Q')v(S)}{v(Q')l(S)}}.$
 13. $m \leftarrow m - 1.$
 14. Return f .
-

Note that line 12 of the algorithm is simplified to $l = (x - x_P)$ and $v = 1$ in the final iteration since $T = -P$ in that case.

Clearly there are $\log_2(n)$ iterations of the main loop of Miller's algorithm and so the doubling operation is executed $\log_2(n)$ times. The number of times the addition operation (lines 10 to 12) is executed is equal to one less than the Hamming weight of n . Hence Miller's algorithm runs in polynomial time.

One way to choose a suitable point S is to choose a random point in $E(K)$. When n is large the algorithm can easily be made deterministic by taking $S = [i]P$, where the binary expansion of i is not a segment of the binary expansion of n (or by taking $S = Q$ if $P \in E(\mathbb{F}_q)$ and $Q \notin E(\mathbb{F}_q)$).

A variation of the algorithm outputs an explicit expression for the function f as a *straight-line program* (i.e., as a product of powers of small polynomials which requires polynomial storage space if it is kept in factored form).

Example: Let $E : y^2 = x^3 + 1$ over $\mathbb{F}_{101}$. Then $\#E(\mathbb{F}_{101}) = 101 + 1 = 2 \cdot 3 \cdot 17$. For $n = 17$ we have $k = 2$. Write $\mathbb{F}_{101^2} = \mathbb{F}_{101}(\theta)$, where $\theta^2 = -2$. Let $P = (87, 61)$ (which has order 17) and let $Q = (48, \theta)$ (which has order 102). Set $D = ([2]Q) - (Q)$. We compute the following values:

i	$f_i(D)$	i	$f_i(D)$
1	1	8	$46 + 18\theta$
2	$52 + 56\theta$	16	$22 + 43\theta$
4	$53 + 3\theta$	17	$74 + 62\theta$

Hence $\langle P, Q \rangle_{17} = 74 + 62\theta$. Raising to the power $(101^2 - 1)/17 = 600$ gives the value $93 + 25\theta \in \mu_{17}$.

IX.9. The MOV/Frey-Rück Attack on the ECDLP

An important application of pairings in elliptic curve cryptography is to transform an instance of the elliptic curve discrete logarithm problem into an instance of a finite field discrete logarithm problem. The motivation for this approach is that there are index-calculus algorithms for the discrete logarithm problem in finite fields which run in subexponential time. An approach using the Weil pairing was given by Menezes, Okamoto and Vanstone [239] while an approach using the Tate pairing was given by Frey and Rück [126].

Let $P \in E(\mathbb{F}_q)$ be of prime order r , coprime to q , and let Q be some multiple of P . The MOV/Frey-Rück attack is as follows:

ALGORITHM IX.2: MOV/Frey–Rück Attack

INPUT: $P, Q \in E(\mathbb{F}_q)$, of prime order r , such that $Q = [\lambda]P$
for some unknown value λ .

OUTPUT: Discrete logarithm λ of Q to the base P .

1. Construct the field $\mathbb{F}_{q^k}$ such that r divides $(q^k - 1)$.
 2. Find⁴ a point $S \in E(\mathbb{F}_{q^k})$ such that $e(P, S) \neq 1$.
 3. $\zeta_1 \leftarrow e(P, S)$.
 4. $\zeta_2 \leftarrow e(Q, S)$.
 5. Find λ such that $\zeta_1^\lambda = \zeta_2$ in $\mathbb{F}_{q^k}^*$ using an index-calculus method (perform linear algebra modulo r).
 6. Return λ .
-

The first steps of this attack require negligible computational resources, but solving the discrete logarithm problem in the finite field $\mathbb{F}_{q^k}$ is non-trivial; the complexity of solving discrete logarithms in $\mathbb{F}_{q^k}^*$ is subexponential in q^k . Hence the attack has exponential complexity in terms of k and this strategy is only effective when k is “small”.

An important contribution of Menezes, Okamoto and Vanstone was the observation that supersingular curves always have small values of k ; see Corollary IX.21. There are also non-supersingular curves which are vulnerable to this attack. For example, for every prime power $q = p^a$ ($p > 2$) there are elliptic curves E over $\mathbb{F}_q$ with $q - 1$ points and the reduction in this case requires no extension of the ground field.

IX.10. Supersingular Elliptic Curves

We first give the definition of supersingularity (see Silverman [307]).

DEFINITION IX.18. *Let E be an elliptic curve over a field $\mathbb{F}_q$, where q is a power of p . Then E is supersingular if one of the following equivalent conditions holds:*

1. $\#E(\mathbb{F}_q) \equiv 1 \pmod{p}$ (equivalently, $\#E(\mathbb{F}_q) = q + 1 - t$ where $p|t$).
2. E has no points of order p over $\overline{\mathbb{F}}_q$.
3. The endomorphism ring of E over $\overline{\mathbb{F}}_q$ is non-commutative (more precisely, is an order in a quaternion algebra).

If E is not supersingular, then it is called ordinary.

Every supersingular elliptic curve in characteristic p has $j(E) \in \mathbb{F}_{p^2}$ and so is isomorphic over $\overline{\mathbb{F}}_p$ to a curve defined over $\mathbb{F}_{p^2}$. But, in general, there are twists defined over $\mathbb{F}_{p^n}$ which are not isomorphic over $\mathbb{F}_{p^n}$ to curves defined over $\mathbb{F}_{p^2}$.

⁴This step is trivial in practice; a random point satisfies the condition with overwhelming probability.

We will now classify the embedding degrees for supersingular elliptic curves. First we need a result of Waterhouse.

THEOREM IX.19. (Waterhouse [344]) *Let p be a prime and let $a \in \mathbb{N}$. Define*

$$T = \{p^a + 1 - \#E(\mathbb{F}_{p^a}) : E \text{ an elliptic curve over } \mathbb{F}_{p^a}\}.$$

Then T equals the set of all $t \in \mathbb{Z}$ with $|t| \leq 2\sqrt{p^a}$ which satisfy one of the following conditions:

1. $\gcd(t, p) = 1$.
2. If a is even, then $t = \pm 2p^{a/2}$.
3. If a is even and $p \not\equiv 1 \pmod{3}$, then $t = \pm p^{a/2}$.
4. If a is odd and $p = 2, 3$, then $t = \pm p^{(a+1)/2}$.
5. If (a is odd) or (a is even and $p \not\equiv 1 \pmod{4}$), then $t = 0$.

The curves corresponding to condition 1 are ordinary while the others are supersingular.

THEOREM IX.20. *The following table lists all possibilities for embedding degree and group structure for supersingular elliptic curves over $\mathbb{F}_q$.*

k	q	$\#E(\mathbb{F}_q)$	Group structure of $E(\mathbb{F}_{q^k})$
1	p^{2b}	$q \pm 2\sqrt{q} + 1$	$(\mathbb{Z}/(\sqrt{q} \pm 1)\mathbb{Z})^2$
2	(5)	$q + 1$	$(\mathbb{Z}/(q + 1)\mathbb{Z})^2$
3	(3)	$q + \sqrt{q} + 1$	$(\mathbb{Z}/(q^{3/2} - 1)\mathbb{Z})^2$
3	(3)	$q - \sqrt{q} + 1$	$(\mathbb{Z}/(q^{3/2} + 1)\mathbb{Z})^2$
4	2^{2b+1}	$q \pm \sqrt{2q} + 1$	$(\mathbb{Z}/(q^2 + 1)\mathbb{Z})^2$
6	3^{2b+1}	$q \pm \sqrt{3q} + 1$	$(\mathbb{Z}/(q^3 + 1)\mathbb{Z})^2$

In the above table the numbers (3) and (5) in the q column correspond to cases 3 and 5 of Theorem IX.19 (in case 3, q is a square). In the first $k = 3$ case the MOV/Frey–Rück attack maps the discrete logarithm into the finite field $\mathbb{F}_{q^{3/2}}$ rather than $\mathbb{F}_{q^3}$.

PROOF. The proof follows from Theorem IX.19 and [320]. Let $q = p^a$. Since E is supersingular, the number of points on E over $\mathbb{F}_q$ is given by $q + 1 - t$, where t satisfies one of the conditions 2 to 5.

The characteristic polynomial of the q -power Frobenius map φ on E over $\mathbb{F}_q$ is given by $P(T) = T^2 - tT + q$. If we factor $P(T) = (T - \alpha)(T - \bar{\alpha})$ over $\mathbb{C}$, then the characteristic polynomial of the q^k -power Frobenius map on E is $(T - \alpha^k)(T - \bar{\alpha}^k)$ and $\#E(\mathbb{F}_{q^k}) = q^k + 1 - \alpha^k - \bar{\alpha}^k$. Proposition 2 of [320] shows that if the characteristic polynomial of the q^k -power Frobenius map is $(T - \alpha^k)^2$, where $\alpha^k \in \mathbb{Z}$, then the group structure of $E(\mathbb{F}_{q^k})$ is $(\mathbb{Z}/|1 - \alpha^k|\mathbb{Z})^2$.

• **Case $t = \pm 2\sqrt{q}$:**

In this case q is a square and the characteristic polynomial of Frobenius is $P(T) = T^2 \pm 2\sqrt{q}T + q = (T \pm \sqrt{q})^2$. Since $(\sqrt{q} + 1)(\sqrt{q} - 1) = (q - 1)$ the embedding degree is $k = 1$.

• **Case $t = \pm\sqrt{q}$:**

In this case q is a square and $\#E(\mathbb{F}_q) = (q \pm \sqrt{q} + 1)$. We have $(q \pm \sqrt{q} + 1)(\sqrt{q} \mp 1) = q^{3/2} \mp 1$ and $(q^{3/2} - 1)(q^{3/2} + 1) = (q^3 - 1)$ and so $k = 3$. In the case of $t = +\sqrt{q}$ we really have $\#E(\mathbb{F}_q)$ divides $(q^{3/2} - 1)$ and so “ $k = 3/2$ ”. The characteristic polynomial of the Frobenius map over $\mathbb{F}_{q^3}$ is $P(T) = (T \pm q^{3/2})^2$.

• **Case $t = \pm p^{(a+1)/2}$:**

In the case $p = 2$ we have $\#E(\mathbb{F}_q) = (2^a \pm 2^{(a+1)/2} + 1)$. The characteristic polynomial of Frobenius is $T^2 \pm 2^{(a+1)/2}T + 2^a$ and its roots are

$$\alpha, \bar{\alpha} = 2^{a/2} \left(\frac{\mp 1 \pm i}{\sqrt{2}} \right).$$

We compute that $\alpha^4 = \bar{\alpha}^4 = -2^{2a}$ and so the characteristic polynomial of Frobenius on E over $\mathbb{F}_{q^4}$ is $(T + q^2)^2$. Hence $[1 + q^2]P = \mathcal{O}$ for all $P \in E(\mathbb{F}_{q^4})$ and so $E(\mathbb{F}_{q^4})$ has group structure $(\mathbb{Z}/(q^2 + 1)\mathbb{Z})^2$. Since $(q + \sqrt{2q} + 1)(q - \sqrt{2q} + 1) = (q^2 + 1)$ divides $(q^4 - 1)$ we deduce that $k = 4$.

In the case $p = 3$ we have $\#E(\mathbb{F}_q) = (q \pm \sqrt{3q} + 1)$. The characteristic polynomial of Frobenius has roots

$$\alpha, \bar{\alpha} = 3^{a/2} \left(\frac{\mp \sqrt{3} \pm i}{2} \right).$$

One sees that $\alpha^6 = \bar{\alpha}^6 = -3^{3a}$ and so the characteristic polynomial of Frobenius on E over $\mathbb{F}_{q^6}$ is $(T + q^3)^2$. Since $(q + 1)(q + \sqrt{3q} + 1)(q - \sqrt{3q} + 1) = (q + 1)(q^2 - q + 1) = (q^3 + 1)$ the result follows.

• **Case $t = 0$:**

In this case $\#E(\mathbb{F}_q) = (q + 1)$ and n divides $(q - 1)(q + 1) = (q^2 - 1)$ and so $k = 2$. The characteristic polynomial of Frobenius over $\mathbb{F}_{p^2}$ is $P(T) = (T + q)^2$.

□

COROLLARY IX.21. (*Menezes, Okamoto and Vanstone [239]*) *Supersingular elliptic curves have $k \leq 6$.*

The table in Theorem IX.20 shows us that when E is supersingular and $\#E(\mathbb{F}_q)$ has a large prime divisor r then there are no difficulties when comparing the groups $E(\mathbb{F}_{q^k})[r]$ and $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$.

THEOREM IX.22. *Let E be a supersingular elliptic curve over $\mathbb{F}_q$ with embedding degree k and let r be a prime dividing $\#E(\mathbb{F}_q)$ with $r > 4\sqrt{q}$. Then $r^2 \parallel \#E(\mathbb{F}_{q^k})$ and $E(\mathbb{F}_{q^k})[r] \cap rE(\mathbb{F}_{q^k}) = \{\mathcal{O}\}$.*

PROOF. If $r > 4\sqrt{q}$, then $r \parallel \#E(\mathbb{F}_q)$. Furthermore, $\#E(\mathbb{F}_q)$ is the unique integer in the Hasse interval which is divisible by r . Since the group orders

specified in Theorem IX.20 are all products of integers in the Hasse interval, it follows that $r^2 \parallel \#E(\mathbb{F}_{q^k})$ and that the full r -torsion is defined over $\mathbb{F}_{q^k}$. $\square$

In traditional cryptographic applications (i.e., where pairings are not used) one usually avoids supersingular curves. This is easily achieved. For example, in characteristic two, none of the elliptic curves $y^2 + xy = x^3 + ax^2 + b$ are supersingular. Criteria for detecting/avoiding supersingular curves of higher genus are given in [130].

IX.11. Applications and Computational Problems from Pairings

In this section we consider some computational problems related to pairings. Applications to cryptography will be discussed in Chapter X.

IX.11.1. Deducing Group Structure. The Weil pairing can be used to determine the group structure of an elliptic curve. The key fact is Corollary IX.11, which gives a test for whether $Q \in \langle P \rangle$. Given $N = \#E(\mathbb{F}_q)$, if N has no square factors, then the group structure of $E(\mathbb{F}_q)$ is isomorphic to $\mathbb{Z}/N\mathbb{Z}$. If r^2 divides N , then there could be a point of order r^2 or two independent points of order r . The Weil pairing shows that one can only have two independent points when r divides $(q - 1)$.

Given the factorization of $\gcd(q - 1, \#E(\mathbb{F}_q))$ the group structure can be determined in random polynomial time using the following algorithm due to Miller [249]. For a rigorous analysis see Kohel and Shparlinski [209].

ALGORITHM IX.3: Miller's Algorithm for Group Structure

INPUT: $E/\mathbb{F}_q$ with $\#E(\mathbb{F}_q) = N_0 N_1$ where $\gcd(N_1, q - 1) = 1$ and all primes dividing N_0 divide $q - 1$.

OUTPUT: Integers r and s such that $E(\mathbb{F}_q) \cong (\mathbb{Z}/r\mathbb{Z}) \times (\mathbb{Z}/s\mathbb{Z})$ as a group.

1. $r \leftarrow 1, s \leftarrow 1$.
 2. While $(rs \neq N_0)$ do:
 3. Choose random points $P', Q' \in E(\mathbb{F}_q)$.
 4. $P \leftarrow [N_1]P', Q \leftarrow [N_1]Q'$.
 5. Find the exact orders m and n of P and Q .
 6. $r \leftarrow \text{lcm}(m, n)$.
 7. $\alpha \leftarrow e_r(P, Q)$.
 8. Let s be the exact order of α in $\mu_r \subseteq \mathbb{F}_q^*$.
 9. Return s and rN_1 .
-

IX.11.2. Separating DH and DDH. An important computational problem in cryptography is the elliptic curve decision Diffie–Hellman problem (see [34]). Let E be an elliptic curve over a field K and let r be a prime.

Decision Diffie–Hellman problem (ECDDH):⁵ Given points P_1, P_2, P_3 and P_4 in $E(K)[r]$ such that $P_2 = [\lambda]P_1$ for some λ , determine whether

$$P_4 = [\lambda]P_3.$$

In certain cases this problem can be solved using pairings as follows. If $e(P_1, P_3) \neq 1$, it is enough to test whether

$$e(P_1, P_4) = e(P_2, P_3).$$

Joux and Nguyen [186] have constructed elliptic curves (both supersingular and ordinary) such that the decision Diffie–Hellman problem can be solved in polynomial time using a pairing, but the (computational) Diffie–Hellman problem is provably as hard as solving the discrete logarithm problem. This suggests that ECDDH really is an easier problem than ECCDH.

IX.11.3. Bilinear Diffie–Hellman Problem. Many of the new applications of pairings in cryptography depend for their security on the difficulty of the following problem which was first stated by Boneh and Franklin [39].

Bilinear Diffie–Hellman problem (BDH): Given $P, Q, P_1 = [a]P$ and $P_2 = [b]P$ such that $e(P, Q) \neq 1$, compute

$$e([ab]P, Q).$$

(This is often written in the special case $Q = \psi([c]P)$, where ψ is a distortion map.)

LEMMA IX.23. *The BDH problem is no harder than either the elliptic curve Diffie–Hellman problem (ECDHP) or the finite field Diffie–Hellman problem.*

PROOF. Given a DH oracle on E we run it on input (P, P_1, P_2) to obtain $[ab]P$ and then compute the value $e([ab]P, Q)$.

Given a DH oracle for the finite field we compute $h_1 = e(P, Q), h_2 = e(P_1, Q) = h_1^a$ and $h_3 = e(P_2, Q) = h_1^b$ and run the oracle on input (h_1, h_2, h_3) . The result is $h_1^{ab} = e([ab]P, Q)$. $\square$

One can also define a decision problem (see Joux [184]).

Decision Bilinear Diffie–Hellman problem (DBDH) : Given P, Q such that $e(P, Q) \neq 1$, and given $P_1 = [a]P, P_2 = [b]P$ and g test whether

$$g \stackrel{?}{=} e([ab]P, Q).$$

The methods of Lemma IX.23 show that the DBDH problem is not harder than the decision Diffie–Hellman problem in the finite field.

⁵Our statement of DDH is very general; the authors of [42] call this co-DDH.

There are many other computational problems relating to pairings (see Joux [184]). One interesting result is the following.

THEOREM IX.24. (*Verheul [335]*) *Let G and H be cyclic groups of prime order r . Let $e : G \times G \rightarrow H$ be a non-degenerate pairing where $H \subset \mathbb{F}_{q^k}^*$ is the image subgroup. If there is an efficiently computable group homomorphism from H to G , then the Diffie–Hellman problem in G and H can be efficiently solved.*

PROOF. (Sketch) Denote by $\phi : H \rightarrow G$ the homomorphism. Let $P, P_2 = [a]P, P_3 = [b]P$ be the input Diffie–Hellman problem and define $e(P, P) = h$. If $\phi(h) = P$, then the problem is trivial: $[ab]P = \phi(e(P_2, P_3))$.

Hence define c to be such that $\phi(h) = [c]P$. Verheul gives a variation on the double-and-add method which computes $Q = [c^{-2}]P$ as follows. Since $c^{-2} \equiv c^{r-3} \pmod{r}$, we want to compute $[c^{r-3}]P$. Given $[c^n]P$ one can compute $\phi(e(P, [c^n]P)) = [c^{n+1}]P$ and $\phi(e([c^n]P, [c^n]P)) = [c^{2n+1}]P$.

One then computes $[ab]P = \phi(e(\phi([c^{-2}]P, P_2), P_3))$. $\square$

IX.12. Parameter Sizes and Implementation Considerations

The hardness of the BDH depends on the hardness of the DH problems both on the elliptic curve $E(\mathbb{F}_q)$ and in the finite field $\mathbb{F}_{q^k}$. For most cryptographic applications based on pairings we therefore need to be working in a subgroup of $E(\mathbb{F}_q)$ of sufficiently large prime order r and we need to be working with a sufficiently large finite field. For current minimum levels of security we require $r > 2^{160}$ and $q^k > 2^{1024}$.

The efficiency depends on the particular scheme, but in general smaller values for q means that arithmetic on the elliptic curve $E(\mathbb{F}_q)$ is faster and transmission of elliptic curve points in $E(\mathbb{F}_q)$ requires less bandwidth. Hence we tend to want to keep q as small as possible and to gain our security from larger values for k . A popular choice is to work with points in $E(\mathbb{F}_q)$, where $q \approx 2^{170}$, and to have a curve with embedding degree $k = 6$ so that $q^k \approx 2^{1024}$.

The two main issues for implementing cryptographic applications are

1. How to find suitable elliptic curves.
2. How to compute $e(P, Q)$ quickly.

We deal with these issues in the next two sections.

In the future, key sizes will have to grow to stay ahead of the increased computational power available to adversaries. There is an asymmetry with the BDH problem which is worth mentioning. On the elliptic curve side, to double the time taken to attack the system one adds an extra two bits to the size of the prime r . On the finite field side, one has to add more than 2 bits to double the security. For example, if elliptic curve key sizes increased by 50% to 240 bits then the finite field key sizes would at least double to 2048 bits. This issue means that embedding degrees of size larger than 6 may be useful in the future (see Subsection IX.15.2).

IX.13. Suitable Supersingular Elliptic Curves

Supersingular curves are suitable for pairing-based cryptosystems since it is possible to have $k = 2, 3, 4$ and 6. Table IX.1 contains the most popular supersingular elliptic curves available for various fields.

TABLE IX.1. Popular Supersingular Elliptic Curves

k	Elliptic curve data
2	$E : y^2 = x^3 + a$ over $\mathbb{F}_p$, where $p \equiv 2 \pmod{3}$ $\#E(\mathbb{F}_p) = p + 1$ Distortion map $(x, y) \mapsto (\zeta_3 x, y)$, where $\zeta_3^3 = 1$.
2	$y^2 = x^3 + x$ over $\mathbb{F}_p$, where $p \equiv 3 \pmod{4}$ $\#E(\mathbb{F}_p) = p + 1$. Distortion map $(x, y) \mapsto (-x, iy)$, where $i^2 = -1$.
3	$E : y^2 = x^3 + a$ over $\mathbb{F}_{p^2}$, where $p \equiv 5 \pmod{6}$ and $a \in \mathbb{F}_{p^2}$, $a \notin \mathbb{F}_p$ is a square which is not a cube. $\#E(\mathbb{F}_{p^2}) = p^2 - p + 1$. Distortion map $(x, y) \mapsto (x^p / (\gamma a^{(p-2)/3}), y^p / a^{(p-1)/2})$, where $\gamma \in \mathbb{F}_{p^6}$ satisfies $\gamma^3 = a$.
4	$E_i : y^2 + y = x^3 + x + a_i$ over $\mathbb{F}_2$, where $a_1 = 0$ and $a_2 = 1$. $\#E_i(\mathbb{F}_{2^l}) = 2^l \pm 2^{(l+1)/2} + 1$ (l odd) Distortion map $(x, y) \mapsto (u^2 x + s^2, y + u^2 s x + s)$, where $u \in \mathbb{F}_{2^2}$ and $s \in \mathbb{F}_{2^4}$ satisfy $u^2 + u + 1 = 0$ and $s^2 + (u + 1)s + 1 = 0$.
6	$E_i : y^2 = x^3 - x + a_i$ over $\mathbb{F}_3$, where $a_1 = 1$ and $a_2 = -1$. $\#E_i(\mathbb{F}_{3^l}) = 3^l \pm 3^{(l+1)/2} + 1$ (l odd). Distortion map $(x, y) \mapsto (\alpha - x, iy)$, where $i \in \mathbb{F}_{3^2}$ and $\alpha \in \mathbb{F}_{3^3}$ satisfy $i^2 = -1$ and $\alpha^3 - \alpha - a_i = 0$.

In the cases $k = 2, 3$ we have chosen complex multiplication (CM) curves with discriminant $D = -3$ and $D = -4$. One can use other CM discriminants to get $k = 2, 3$ (for instance, Examples 5 and 6 of [138] in characteristic p when $(\frac{-7}{p}) = -1$ or $(\frac{-2}{p}) = -1$). As we have seen, embedding degree 6 is the largest which can be achieved using supersingular elliptic curves. We discuss methods to obtain larger values of k in Section IX.15.

Note that in the characteristic 2 example the distortion map acts as a cube root of -1 while the characteristic 3 example acts as a square root of -1 . It follows that in both cases the elliptic curve has an endomorphism ring isomorphic to an order in a quaternion algebra which contains $\mathbb{Z}[i]$ or $\mathbb{Z}[\zeta_3]$ as a subring.

One problem with the examples in characteristics 2 and 3 is that there are only a small number of curves and fields to choose from. Hence, there is an element of luck in the search for a suitable large prime factor and our choice of parameters is not very flexible. No such problems arise when working over $\mathbb{F}_p$. Two nice examples in characteristic three are over $\mathbb{F}_{3^{163}}$. We find that

$\#E_1(\mathbb{F}_{3^{163}})$ is 7 times a 256-bit prime and that $\#E_2(\mathbb{F}_{3^{163}})$ is equal to a 259-bit prime (see [130] for details).

Another consideration for the case of a small characteristic is Copper-smith's algorithm [86] for discrete logarithms in finite fields. This algorithm was originally described in the case of characteristic two but it can be generalized to other fields of small characteristic. It is more efficient than the general methods for taking discrete logarithms in finite fields. So one is obliged to work with larger fields to get equivalent security to the case of $\mathbb{F}_p$.

IX.14. Efficient Computation of the Tate Pairing

We turn to the issue of efficient computation of the Tate pairing. We consider the case where E is an elliptic curve over $\mathbb{F}_q$ with $k > 1$ (typically $k = 4$ or 6) and we are computing $\langle P, Q \rangle_n^{(q^k-1)/n}$ with $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^k})$.

The field $\mathbb{F}_{q^k}$ is much larger than the field $\mathbb{F}_q$. Hence we perform as many operations in $E(\mathbb{F}_q)$ as possible. The best approach is to represent $\mathbb{F}_{q^k}$ as an extension of $\mathbb{F}_q$ with a convenient basis.

The most important implementation techniques are the following. We refer to Galbraith, Harrison and Soldera [133] and Barreto, Kim, Lynn and Scott [16, 18] for further details.

1. If possible, work in a small subgroup ($n \approx 2^{160}$) of $E(\mathbb{F}_q)$ and/or choose n to have low Hamming weight (in signed binary representation or signed ternary representation as appropriate).
2. Work over the small field $\mathbb{F}_q$ wherever possible. In particular, the function f with divisor $(f) = n(P) - n(\mathcal{O})$ is defined over $\mathbb{F}_q$ and so the lines l and v in Miller's algorithm are all defined over $\mathbb{F}_q$. Furthermore, choose the auxiliary point S in Miller's algorithm to lie in $E(\mathbb{F}_q)$. This significantly reduces the number of computations (though see point 6 below, where S is omitted entirely when $k > 1$).
3. Implement arithmetic in $\mathbb{F}_q$ and $\mathbb{F}_{q^k}$ as efficiently as possible. This is particularly relevant in characteristic three.
4. Avoid divisions in $\mathbb{F}_{q^k}$. This is done by replacing the variable f in Miller's algorithm by f_1/f_2 . Hence, for example, line 8 of Miller's algorithm becomes

$$f_1 = f_1^2 l(Q') v(S), \quad f_2 = f_2^2 v(Q') l(S).$$

A single division is performed at the end of the algorithm.

5. Perform final exponentiation efficiently. In particular, take advantage of the linearity of the q th power Frobenius map for exponents of special form. Sometimes several of these exponentiations can be performed in one. For example, to test whether $e(P, Q) \stackrel{?}{=} e(R, S)$, one can test whether $(\langle P, Q \rangle_n / \langle R, S \rangle_n)^{(q^k-1)/n} \stackrel{?}{=} 1$.

6. If $k > 1$ and n is prime, then n does not divide $(q-1)$ and so all elements of $\mathbb{F}_q^*$ map to 1 when raised to the power $(q^k - 1)/n$. Hence we can ignore all terms in Miller's algorithm which give rise to an element of $\mathbb{F}_q^*$. For example, since l, v and S are defined over $\mathbb{F}_q$, we need not compute $l(S)$ and $v(S)$ at all. Line 8 of Miller's algorithm becomes

$$f_1 = f_1^2 l(Q'), \quad f_2 = f_2^2 v(Q').$$

Indeed, we may now take $Q' = Q$ (see Theorem 1 of Barreto et al. [18]).

7. If we are using non-rational endomorphisms of a suitable form, then all denominators in Miller's algorithm can be removed (see Barreto et al. [16, 18]).

Two techniques which are often used to speed-up point exponentiation for elliptic curves are window methods (see [ECC, Section IV.2.3]) and projective coordinates ([ECC, Section IV.1]). We now argue that these methods are not useful in the case of the Tate pairing.

Firstly, when n has low Hamming weight (which is a common case) there is no advantage from using window methods. Now suppose n is arbitrary and suppose we use windows of length w . We precompute a table of points $[d]P$ (where $-2^{w-1} < d \leq 2^{w-1}$) and corresponding function values $f_{d,1}/f_{d,2}$ which record the value at the divisor D of the function f_d . The key observation is that, when performing an addition of $[d]P$ rather than P , line 12 of Miller's algorithm changes from

$$f_1 = f_1 l(Q'), \quad f_2 = f_2 v(Q') \quad \text{to} \quad f_1 = f_1 f_{d,1} l(Q'), \quad f_2 = f_2 f_{d,2} v(Q').$$

Hence, as well as the significant initial cost of setting up the table for the window method, there is an increase from two to four $\mathbb{F}_{q^k}$ -multiplications in each addition stage. If we compute the division during the precomputation (i.e., $f_{d,2} = 1$), then we still require three $\mathbb{F}_{q^k}$ -multiplications at the expense of greater cost during the precomputation. One can therefore show that window methods do not provide a speedup in this setting.

If group orders without low Hamming weight are used, then the methods of Eisenträger, Lauter and Montgomery [114] for performing $[2]P + Q$ in one step give a significant efficiency improvement.

There is no need to use projective coordinates for the points in $E(\mathbb{F}_{q^k})$ as we have already bundled all the $\mathbb{F}_{q^k}$ divisions to a single division at the end. One can use projective coordinates for the operations in $E(\mathbb{F}_q)$. The performance analysis depends on the relative costs of inversion and multiplication in $\mathbb{F}_q$. This idea was examined by Izu and Takagi [180], but their analysis is misleading. Using projective coordinates does not reduce the number of $\mathbb{F}_{q^k}$ operations and experiments show that affine coordinates are faster.

In equation (IX.1) we saw that the Weil pairing can be computed by essentially two Tate pairing computations. Hence one might expect that the Tate pairing is about twice as fast as the Weil pairing (except for the extra

exponentiation, which is required for the Tate pairing). In fact, since we usually have $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^k})$, the cost of computing $\langle Q, P \rangle_n$ is considerably higher than the cost of computing $\langle P, Q \rangle_n$. Hence the Tate pairing is more than twice as fast as the Weil pairing.

IX.14.1. Doubling in Characteristic Two. Let $E : y^2 + y = x^3 + x + a$ be a supersingular curve in characteristic 2 with embedding degree $k = 4$. As is well known, doubling of points on E can be performed without divisions.

Let $P = (x_1, y_1)$. Then the tangent line to E at P has slope $\lambda = (x_1^2 + 1)$ and has equation $l : y = \lambda(x - x_1) + y_1$. The third point of intersection has x -coordinate $x_2 = \lambda^2 = x_1^4 + 1$. The vertical line has equation $x - x_2 = 0$ and $[2](x_1, y_1) = (x_2, y_2)$, where $y_2 = 1 + \lambda(x_2 + x_1) + y_1$. Hence the point (x_2, y_2) and the equations of the lines l and v can be computed from (x_1, y_1) by two applications of the squaring map (easy in characteristic two), one multiplication and some additions.

The equation $[2](x_1, y_1) = (x_1^4 + 1, y_1^4 + x_1^4)$ (if $a \in \mathbb{F}_2$) shows a connection between doubling and the action of Frobenius squared. The characteristic polynomial of Frobenius in this example is $T^2 \pm 2T + 2$. Since

$$(T^2 + 2T + 2)(T^2 - 2T + 2) = (T^4 + 4)$$

we have $[4] = -\varphi^4$, where φ is the 2-power Frobenius. This gives the formula

$$[4](x_1, y_1) = (x_1^{16}, y_1^{16} + 1)$$

which is clearly a consequence of our doubling formula.

IX.14.2. Tripling in Characteristic Three. Suppose $P = (x_1, y_1)$ is a point on

$$E : y^2 = x^3 + a_4x + a_6$$

over $\mathbb{F}_{3^m}$. The tangent to E at P has slope $\lambda_2 = 1/y_1$ and

$$[2]P = (x_2, y_2) = (\lambda_2^2 + x_1, -\lambda_2^3 - y_1).$$

The line between (x_1, y_1) and $[2](x_1, y_1)$ has slope $\lambda_3 = y_1^3 - \lambda_2$. Putting this together, if $a_4, a_6 \in \mathbb{F}_3$, then

$$[3](x_1, y_1) = (x_1^9 + a_6(1 - a_4), -y_1^9).$$

So tripling is division-free, although one division is required⁶ to compute the equations of the lines l and v . Note that there is a connection between tripling and Frobenius squared.

Since tripling is so efficient it makes sense to utilize a base three Miller algorithm in characteristic three. Furthermore, as the next subsection shows, one can use group orders with low Hamming weight in base three which makes the base three Miller's algorithm very efficient.

⁶This division can usually be avoided by homogenising, since it contributes an element of $\mathbb{F}_q$ which is removed during the final exponentiation.

The fact that tripling is easy on certain curves in characteristic three was first noticed by Duursma [110] (Lemma 3.1). Let $P = (a, b)$ be a point on $y^2 = x^3 - x + d$. Then the function $h = b^3y - (a^p - x + d)^{(p+1)/2}$ satisfies

$$(h) = 3(x, y) - 3(\mathcal{O}) + (-3P) - (\mathcal{O}).$$

This has been used by Duursma and Lee [112] to obtain a faster implementation of the Tate pairing in characteristic three.

IX.14.3. Low Hamming Weight Example. Often we are not able to choose the order of P to be a prime of low Hamming weight. For example,

$$\#E_1(\mathbb{F}_{3^{163}}) = N = 3^{163} - 3^{82} + 1 = 7r$$

where r is a prime. The prime r does not have low Hamming weight in base three. So instead of computing $\langle P, Q \rangle_r^{(3^{163}-1)/r}$ we compute $\langle P, Q \rangle_N^{(3^{163}-1)/N}$. By part 1 of Theorem IX.9 the pairing value is the same in both cases.

Note that the exponentiation is to the power

$$((3^{163})^6 - 1)/N = (3^{163 \cdot 4} + 3^{163 \cdot 3} - 3^{163} - 1)(3^{163} + 3^{82} + 1)$$

which also has low Hamming weight. Moreover, this exponentiation can be computed using repeated applications of the Frobenius map, which is very efficient.

IX.15. Using Ordinary Curves

As mentioned in Section IX.12, it would be useful to have families of curves with increasing k .

With supersingular elliptic curves the upper limit is $k = 6$. Rubin and Silverberg [306] have shown how to obtain values of k larger than 6 from a combination of supersingular elliptic curves and Weil descent. Their methods are quite practical, and they give a significant improvement to the BLS short signature scheme (see Section 5.1 of [306] for details). However, the possibilities for k are somewhat limited by efficiency considerations. Another avenue is to use higher genus supersingular curves (see Galbraith [130] and Rubin and Silverberg [306] for details). However, there are severe limitations in the higher genus case too.

Instead we turn to ordinary (i.e., non-supersingular) elliptic curves. There are two problems with ordinary curves in this context.

- As we have seen, there are no distortion maps in this case. Nevertheless, many of the cryptosystems can be easily modified to handle this issue.
- Results of Balasubramanian and Koblitz [12] show that such curves are extremely rare. More precisely, if k is fixed, then the expected number of pairs (q, E) , where q is a prime power in the range $M/2 \leq q \leq M$ and E is an $\mathbb{F}_q$ -isogeny class of elliptic curves over $\mathbb{F}_q$ such that $E(\mathbb{F}_q)$ has a large subgroup with embedding degree k , is $O(M^{1/2+\epsilon})$. Hence we cannot expect to find them by choosing curves at random.

For the remainder of this section we discuss some special methods for constructing ordinary elliptic curves with convenient embedding degrees. All these algorithms are based on the CM method; see [ECC, Chapter VIII]. We begin by showing that degree 3, 4 and 6 can be recovered in the ordinary case, and then later show how to find curves with $k > 6$.

IX.15.1. The MNT Criteria. The first efforts in this direction were obtained by Miyaji, Nakabayashi and Takano [251] (MNT). They showed how to obtain the values $k = 3, 4, 6$ with ordinary elliptic curves.

THEOREM IX.25. ([251]) *Let E be an ordinary elliptic curve over $\mathbb{F}_q$ such that $n = \#E(\mathbb{F}_q) = q + 1 - t$ is prime. Then the following table lists all the possibilities for embedding degree $k = 3, 4, 6$:*

k	q	t
3	$12l^2 - 1$	$-1 \pm 6l$
4	$l^2 + l + 1$	$-l$ or $l + 1$
6	$4l^2 + 1$	$1 \pm 2l$

It is an easy exercise to check that these group orders have the claimed embedding degrees. The harder part is to show that any such curve with prime order has these properties. Note that all these examples produce values for q in quadratic progressions, which agrees with the results of Balasubramanian and Koblitz.

Miyaji, Nakabayashi and Takano developed a method to construct such curves using complex multiplication; see [ECC, Chapter VIII]. As is usual for the CM method, the trick is to choose q and t in such a way that $\Delta = t^2 - 4q$ is a large square times a small discriminant.

We give the details of their method in the case $k = 6$. For $q = 4l^2 + 1$ and $t = 1 \pm 2l$ we find that $t^2 - 4q = -12l^2 \pm 4l - 3$. Hence we must solve

$$-12l^2 \pm 4l - 3 = Dy^2$$

where D is a small CM discriminant. Multiplying by -3 and completing the square gives $-3Dy^2 = (6l \mp 1)^2 + 8$. Hence, we seek solutions (x, y) to the Diophantine equation $x^2 + 3Dy^2 = -8$. This equation is solved by finding a single solution and then combining with solutions found using the continued fraction method to the Pell equation $x^2 + 3Dy^2 = 1$. If a solution has $x \equiv \pm 1 \pmod{6}$, we can compute the corresponding l and test whether $4l^2 + 1$ is prime. Since the Diophantine equation has infinitely many solutions we have a good chance of success.

Example: Take $D = -11$. The first step is to find solutions to

$$x^2 - 33y^2 = -8$$

(note that $(\frac{-8}{3}) = (\frac{-8}{11}) = 1$ so we can hope to succeed). We find an initial solution $(x, y) = (5, 1)$.

Now, to generate infinitely many solutions we find all the solutions to

$$x^2 - 33y^2 = 1$$

using continued fractions. The first three solutions are $(23, 4)$, $(1057, 184)$ and $(48599, 8460)$. The first of these solutions combined with the initial solution $(5, 1)$ gives the solution $(x, y) = (17, 3)$ to the original equation.

Since x is of the form $6l \pm 1$ we obtain $l = 3$ and thus $q = 4l^2 + 1 = 37$, which is prime. The trace is $t = 1 \pm 2l$ so $t = 7$ can be obtained, and this gives a curve with 31 points and the Frobenius has discriminant $t^2 - 4q = -3^2 11$.

The elliptic curve with complex multiplication by $D = -11$ has j -invariant equal to -32768 . Hence we find an equation

$$E : y^2 = x^3 + 22x + 27$$

over $\mathbb{F}_{37}$, which can be checked to have 31 points. One can check that

$$\#E(\mathbb{F}_{37^6}) = 2^6 \cdot 3^4 \cdot 5 \cdot 31^2 \cdot 103.$$

IX.15.2. Obtaining Larger Values of k . We can now consider how to generate curves with larger values for k using the CM method. The goal is to find an elliptic curve E over $\mathbb{F}_q$ with complex multiplication by D such that there is a large prime r dividing $\#E(\mathbb{F}_q) = q + 1 - t$ and such that r divides $(q^k - 1)$ for a suitable value of k . The condition r divides $(q^k - 1)$ does not preclude r dividing $(q^l - 1)$ for some $l < k$; hence we often use the condition r divides $\Phi_k(q)$ where $\Phi_k(x)$ denotes the k th cyclotomic polynomial (see Lang [210] Section VI.3) which is the factor of $(x^k - 1)$ which does not appear as a factor of any polynomial $(x^l - 1)$ with $l < k$.

We can express the requirements by the equations

$$t^2 - 4q = f^2 D \tag{IX.2}$$

$$q + 1 - t \equiv 0 \pmod{r} \tag{IX.3}$$

$$q^k - 1 \equiv 0 \pmod{r}. \tag{IX.4}$$

Equation (IX.3) gives $q \equiv t - 1 \pmod{r}$ and so equation (IX.4) is equivalent to $(t - 1)^k \equiv 1 \pmod{r}$. In other words, $t - 1$ is a k th root of unity modulo r . Equation (IX.2) gives $4q = t^2 - f^2 D$. If $t = 2a$ is even, then f is even and we obtain $q = a^2 - (f/2)^2 D$. If $t = 2a + 1$ is odd, then we obtain $q = a^2 + a + (1 - f^2 D)/4$.

If values for r and k are fixed, then there are only $k - 1$ possible values for t modulo r and, for each of these choices, q is determined modulo r . One can search for prime values of q by searching along a certain arithmetic progression depending on r . This gives rise to the following algorithm (originally proposed by Cocks and Pinch [84]) for generating curves using the CM method with arbitrary embedding degree k and with a large prime factor r of any form.

We give the algorithm in the case $D \equiv 0 \pmod{4}$ (the case $D \equiv 1 \pmod{4}$) is similar). Let $b = f/2$ so that we seek $q = a^2 - b^2 D$. Substituting this

expression and $t = 2a$ into equation (IX.3) gives $(a - 1)^2 - b^2 D \equiv 0 \pmod{r}$. This implies that $b = \pm(a - 1)/\sqrt{D} \pmod{r}$. There are usually several reasonable choices for g, a and b_0 , and the algorithm should be run for all choices to obtain optimal results.

ALGORITHM IX.4: Curve Construction for Arbitrary k

INPUT: $k, r, D \equiv 0 \pmod{4}$ such that r is a prime, D is a square modulo r and k divides $(r - 1)$.

OUTPUT: p, t such that there is a curve with CM by D over $\mathbb{F}_p$ with $p + 1 - t$ points where $r \mid (p + 1 - t)$ and $r \mid (p^k - 1)$.

1. Choose a primitive k th root of unity g in $\mathbb{F}_r$.
 2. Choose an integer $a \equiv 2^{-1}(g + 1) \pmod{r}$.
 3. If $(\gcd(a, D) \neq 1)$, then halt (or choose another g).
 4. Choose an integer $b_0 \equiv \pm(a - 1)/\sqrt{D} \pmod{r}$.
 5. $j \leftarrow 0$.
 6. Do
 7. $p \leftarrow a^2 - D(b_0 + jr)^2$.
 8. $j \leftarrow j + 1$.
 9. Until p is prime.
 10. $t \leftarrow 2a$.
 11. Return p and t .
-

Example: We illustrate the above algorithm in an example. Let $D = -3$, $k = 11$ and $r = 2147483713 = 2^{31} + 2^6 + 1$. Once can check that r is a prime of low Hamming weight such that k divides $(r - 1)$ and $(\frac{-3}{r}) = 1$.

An 11th root of unity modulo r is easily found by taking a random element modulo r and raising it to the power $(r - 1)/11$. We take $g = 865450015$, which gives $a = (g + 1)/2 = 432725008$. At this stage we know that r divides $((2a - 1)^k - 1)$.

Set $b_0 = (a - 1)/\sqrt{D} \equiv 1946126982 \pmod{r}$. Then r divides $((a - 1)^2 - Db_0^2)$ and so, for every number of the form $p = a^2 - D(b_0 + jr)^2$, we have r divides $(p + 1 - 2a)$. For the values $j = 17, 25, 41, 45, \dots$ we find that $a^2 - D(b_0 + jr)^2$ is a prime. Taking $j = 17$ gives $p = 4436167653364218931891$ which has 72 bits.

The elliptic curve $y^2 = x^3 + 1$ modulo p is seen to have $p + 1 - 2a$ points (which is divisible by r) and the ring generated by the Frobenius endomorphism has discriminant $4a^2 - 4p = 4(b_0 + 17r)^2 D$.

The main drawback of this algorithm is that $p > r^2$ (compared with the supersingular case, where we had $p \approx r$). An algorithm with some features similar to the above has been given by Dupont, Enge and Morain [109]. More sophisticated methods for generating curves with embedding degree $k > 6$ have been given by Barreto, Lynn and Scott [17, 18] and Brezing and

Weng [49]. In particular, their results have improved relationships between the sizes of r and p for elliptic curves with embedding degree of special form such as $k = 2^i 3^j$ (e.g., $k = 12$). This problem is still an active area of research.

Appendix: Proof of Weil reciprocity

Before we can give the proof of Weil reciprocity we need some further technical ingredients.

A non-constant rational map $\phi : C_1 \rightarrow C_2$ induces a map $\phi^* : K(C_2) \rightarrow K(C_1)$ on the function fields given by $\phi^* f = f \circ \phi$. We can consider $K(C_1)$ as a finite algebraic extension of $\phi^* K(C_2)$. The *degree* of ϕ is $\deg(\phi) = [K(C_1) : \phi^* K(C_2)]$. The map $\phi_* : K(C_1) \rightarrow K(C_2)$ is defined by $\phi_* f = (\phi^*)^{-1} \circ N_{K(C_1)/\phi^* K(C_2)}(f)$, where $N_{K(C_1)/\phi^* K(C_2)}$ is the usual definition of norm for a finite extension of fields.

Given a non-constant rational map $\phi : C_1 \rightarrow C_2$ and a point P on C_1 we may define the *ramification index* $e_\phi(P)$ in terms of uniformizers (see Silverman [307] II.2). The ramification index satisfies $e_\phi(P) \text{ord}_{\phi(P)}(f) = \text{ord}_P(\phi^* f)$ for any function f on C_2 (see [307] Exercise 2.2). The map ϕ induces maps between the divisors on C_1 and C_2 as follows. We have $\phi^* : \text{Div}(C_2) \rightarrow \text{Div}(C_1)$ induced by $\phi^* : (Q) \mapsto \sum_{P \in \phi^{-1}(Q)} e_\phi(P)(P)$. We have $\phi_* : \text{Div}(C_1) \rightarrow \text{Div}(C_2)$ induced by $\phi_* : (Q) \mapsto (\phi(Q))$.

A non-constant function f on C can be thought of as a non-constant rational map $f : C \rightarrow \mathbb{P}^1$ over K . Given $f : C \rightarrow \mathbb{P}^1$ we have $(f) = f^*((0) - (\infty))$.

LEMMA IX.26. *Let $\phi : C_1 \rightarrow C_2$ be a non-constant rational map, let $f_i : C_i \rightarrow \mathbb{P}^1$ for $i = 1, 2$ be functions and let D_i be divisors on C_i for $i = 1, 2$. We have the following properties:*

1. $\phi^*((f_2)) = (\phi^* f_2)$.
2. $\phi_*((f_1)) = (\phi_* f_1)$.
3. $\phi_* \circ \phi^* = [\deg \phi]$ in $\text{Div}(C_2)$.
4. $f_1(\phi^* D_2) = (\phi_* f_1)(D_2)$.
5. $f_2(\phi_* D_1) = (\phi^* f_2)(D_1)$.

PROOF. For 1, 2 and 3 see Silverman [307, Proposition II.3.6]. For 4 and 5 see Silverman [307, Exercise 2.10]. $\square$

PROOF. (of Weil reciprocity Theorem IX.3) We first prove the result in the case $C = \mathbb{P}^1$. We identify $\mathbb{P}^1(\overline{K})$ with $\{a \in \overline{K}\} \cup \{\infty\}$. A function f on C is a ratio $u(x, z)/v(x, z)$ of two homogeneous polynomials over $\overline{K}$ of the same degree. In practice, we can consider the restriction of f to an affine set, and so write f as a ratio $u(x)/v(x)$, where $u(x), v(x) \in \overline{K}[x]$. If $f = \prod_{i=1}^m (x - a_i)^{n_{a_i}}$, then $(f) = \sum_{i=1}^m n_{a_i}(a_i) - n_\infty(\infty)$, where $n_\infty = \sum_{i=1}^m n_{a_i} = \deg(u(x)) - \deg(v(x))$, and vice versa.

Let f be as above, let $g = \prod_{j=1}^{m'} (x - b_j)^{n_{b_j}}$ and suppose that both (f) and (g) have disjoint support with no points at infinity (in other words, $a_i \neq b_j$ for all i and j and $\sum_{i=1}^m n_{a_i} = \sum_{j=1}^{m'} n_{b_j} = 0$). Then

$$\begin{aligned} f((g)) &= \prod_{j=1}^{m'} \prod_{i=1}^m (b_j - a_i)^{n_{a_i} n_{b_j}} \\ &= (-1)^{(\sum_{i=1}^m \sum_{j=1}^{m'} n_{a_i} n_{b_j})} \prod_{j=1}^{m'} \prod_{i=1}^m (a_i - b_j)^{n_{a_i} n_{b_j}} \\ &= g((f)) \end{aligned}$$

where the sign is 1 since

$$\sum_{i=1}^m \sum_{j=1}^{m'} n_{a_i} n_{b_j} = \left(\sum_{i=1}^m n_{a_i} \right) \left(\sum_{j=1}^{m'} n_{b_j} \right) = 0 \cdot 0 = 0.$$

The case where ∞ appears in the support of either (f) or (g) arises when the degree of the numerator is not equal to the degree of the denominator. We define $(\infty - b_i)/(\infty - b_j) = 1$ and the rest of the proof proceeds as before. This proves the result for $\mathbb{P}^1$.

In the general case let i be the identity function on $\mathbb{P}^1$. Then $(i) = (0) - (\infty)$ and as noted above $(g) = g^*((i))$. We therefore have

$$f((g)) = f(g^*((i))) = (g_* f)((i)).$$

Now, $g_* f$ is a function on $\mathbb{P}^1$ and so, by Weil reciprocity on $\mathbb{P}^1$, we have $(g_* f)((i)) = i((g_* f))$. To complete the proof we note that

$$i((g_* f)) = (g^* i)((f)) = i \circ g((f)) = g((f)).$$

□

CHAPTER X

Cryptography from Pairings

K.G. Paterson

X.1. Introduction

This chapter presents a survey of positive applications of pairings in cryptography. We assume the reader has a basic understanding of concepts from cryptography such as public-key encryption, digital signatures, and key exchange protocols. A solid grounding in the general area of cryptography can be obtained by reading [240].

We will attempt to show how pairings (as described in Chapter IX) have been used to construct a wide range of cryptographic schemes, protocols and infrastructures supporting the use of public-key cryptography. Recent years have seen an explosion of interest in this topic, inspired mostly by three key contributions: Sakai, Ohgishi and Kasahara's early and much overlooked work introducing pairing-based key agreement and signature schemes [284]; Joux's three-party key agreement protocol as presented in [183]; and Boneh and Franklin's identity-based encryption (IBE) scheme built from pairings [38]. The work of Verheul [335] has also been influential because it eases the cryptographic application of pairings. We will give detailed descriptions of these works as the chapter unfolds. To comprehend the rate of increase of research in this area, note that the bibliography of an earlier survey [273] written in mid-2002 contains 28 items, while, at the time of writing in early 2004, Barreto's website [15] lists over 100 research papers.¹

Thus a survey such as this cannot hope to comprehensively cover all of the pairing-based cryptographic research that has been produced. Instead, we focus on presenting the small number of schemes that we consider to be the high points in the area and which are likely to have a significant impact on future research. We provide brief notes on most of the remaining literature and omit some work entirely. We do not emphasise the technical details of security proofs, but we do choose to focus on schemes that are supported by such proofs.

¹A second source for papers on cryptography from pairings is the IACR preprint server at <http://eprint.iacr.org>. Another survey on pairings and cryptography by Joux [184] covers roughly the same topics as this and the previous chapter.

X.1.1. Chapter Plan. In the next two sections we introduce the work of Sakai *et al.* [284], Joux [183] and Boneh and Franklin [38]. Then in Section X.4, we consider various types of signature schemes derived from pairings. Section X.5 is concerned with further developments of the IBE scheme of [38] in the areas of hierarchical identity-based cryptography, intrusion-resilient cryptography and related topics. Section X.6 considers how the key agreement protocols of [284, 183] have been extended. In the penultimate section, Section X.7, we look more closely at identity-based cryptography and examine the impact that pairings have had on infrastructures supporting the use of public-key cryptography. We also look at a variety of trials and implementations of pairing-based cryptography. We draw to a close with a look towards the future in Section X.8.

X.1.2. Pairings as Black Boxes. In this chapter we will largely treat pairings as “black boxes,” by which we mean that we will not be particularly interested in how the pairings can be selected, computed and so on. Rather we will treat them as abstract mappings on groups. Naturally, Chapter IX is the place to look for the details on these issues. The reason to do this is so that we can concentrate on the general cryptographic principles behind the schemes and systems we study, without being distracted by the implementation details. It does occasionally help to look more closely at the pairings, however. For one thing, the availability of easily computable pairings over suitably “compact” groups and curves is key to the utility of some of the pairing-based proposals that we study. And of course the real-world security of any proposal will depend critically on the actual curves and pairings selected to implement that proposal. It would be inappropriate in a chapter on applications in cryptography to completely ignore these issues of efficiency and security. So we will “open the box” whenever necessary.

Let us do so now, in order to re-iterate some notation from the previous chapter and to establish some of the basics for this chapter. We recall the basic properties of a pairing $e : G_1 \times G_2 \rightarrow G_3$ from Section IX.1. In brief, e is a bilinear and non-degenerate map and will be derived from a Tate or Weil pairing on an elliptic curve $E(\mathbb{F}_q)$. In cryptographic applications of pairings, it is usually more convenient to work with a single subgroup G_1 of $E(\mathbb{F}_q)$ having prime order r and generator P as input to the pairing, instead of two groups G_1 and G_2 . For this reason, many of the schemes and systems we study were originally proposed in the context of a “self-pairing” as described in Section IX.7. To ensure that the cryptographic schemes are not completely trivial, it is then important that $e(P, P) \neq 1$. The distortion maps of Verheul [335] are particularly helpful in ensuring that these conditions can be met for supersingular curves.

As in Section IX.7.3, we assume that $E(\mathbb{F}_q)$ is a supersingular elliptic curve with $r \mid \#E(\mathbb{F}_q)$ for some prime r . We write $k > 1$ for the embedding degree for E and r and assume that $E(\mathbb{F}_{q^k})$ has no points of order r^2 . As usual,

we write $e(Q, R) = \langle Q, R \rangle_r^{(q^k-1)/r} \in \mathbb{F}_{q^k}$ for $Q \in E(\mathbb{F}_q)[r]$ and $R \in E(\mathbb{F}_{q^k})$. We then let φ denote a non-rational endomorphism of E (a distortion map). Suitable maps φ are defined in Table IX.1. We put $G_1 = \langle P \rangle$, where P is any non-zero point in $E(\mathbb{F}_q)[r]$ and $G_3 = \mathbb{F}_{q^k}^* / (\mathbb{F}_{q^k}^*)^r$. We then write $\hat{e}$ for the map from $G_1 \times G_1$ to G_3 defined by:

$$\hat{e}(Q, R) = e(Q, \varphi(R)).$$

The function $\hat{e}$ is called a *modified* pairing. As a consequence of its derivation from the pairing e and distortion map φ , it has the following properties:

Bilinearity: For all $Q, Q', R, R' \in G_1$, we have

$$\hat{e}(Q + Q', R) = \hat{e}(Q, R) \cdot \hat{e}(Q', R)$$

and

$$\hat{e}(Q, R + R') = \hat{e}(Q, R) \cdot \hat{e}(Q, R').$$

Symmetry: For all $Q, R \in G_1$, we have

$$\hat{e}(Q, R) = \hat{e}(R, Q).$$

Non-degeneracy: We have

$$\hat{e}(P, P) \neq 1.$$

Hence we have: $\hat{e}(Q, P) \neq 1$ for all $Q \in G_1$, $Q \neq \mathcal{O}$ and $\hat{e}(P, R) \neq 1$ for all $R \in G_1$, $R \neq \mathcal{O}$.

Although our notation inherited from the previous chapter suggests that the map $\hat{e}$ must be derived from the Tate pairing, this need not be the case. The Weil pairing can also be used. However, as Chapter IX spells out, the Tate pairing is usually a better choice from an implementation perspective.

Relying on distortion maps in this way limits us to using supersingular curves. There may be good implementation or security reasons for working with curves other than these, again as Chapter IX makes clear. (In particular, special purpose algorithms [2, 3, 86, 185] can be applied to solve the discrete logarithm problem in $\mathbb{F}_{q^k}$ when E is one of the supersingular curves over a field of characteristic 2 or 3 in Table IX.1. This may mean that larger parameters than at first appears must be chosen to obtain the required security levels.) Most of the cryptographic schemes that were originally defined in the self-pairing setting can be adapted to operate with ordinary curves and unmodified pairings, at the cost of some minor inconvenience (and sometimes a loss of bandwidth efficiency). We will encounter situations where ordinary curves are in fact to be preferred. Moreover, we will present some schemes using the language of self-pairings that were originally defined using unmodified pairings. We will note in the text where this is the case.

We can summarize the above digression into some of the technicalities of pairings as follows. By carefully selecting an elliptic curve $E(\mathbb{F}_q)$, we can obtain a symmetric, bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_3$ with the property that $\hat{e}(P, P) \neq 1$. Here, P of prime order r on $E(\mathbb{F}_q)$ generates G_1 and

G_3 is a subgroup of $\mathbb{F}_{q^k}$ for some small k . When parameters $\langle G_1, G_3, \hat{e} \rangle$ are appropriately selected, we also have the following properties:

Efficiency: The computation of $\hat{e}$ can be made relatively efficient (equivalent perhaps to a few point multiplications on $E(\mathbb{F}_q)$). Elements of G_1 and G_3 have relatively compact descriptions as bit-strings, and arithmetic in these groups can be efficiently implemented.

Security: The bilinear Diffie–Hellman problem and the decision bilinear Diffie–Hellman problem are both computationally hard.²

X.2. Key Distribution Schemes

In this section we review the work of Sakai et al. [284] and Joux [183] on key distribution schemes built from pairings. These papers paved the way for Boneh and Franklin’s identity-based encryption scheme, the subject of Section X.3. Note that both papers considered only unmodified pairings. We have translated their schemes into the self-pairing setting in our presentation.

X.2.1. Identity-Based Non-Interactive Key Distribution. Key distribution is one of the most basic problems in cryptography. For example, frequently refreshed, random keys are needed for symmetric encryption algorithms and MACs to create confidential and integrity-protected channels. Consider the situation of two parties A and B who want to compute a shared key K_{AB} but cannot afford to engage in a Diffie–Hellman protocol (perhaps one of them is initially offline or they cannot afford the communications overhead of an interactive protocol).

Sakai et al. [284] proposed a pairing-based solution to this problem of constructing a non-interactive key distribution scheme (NIKDS). An important and interesting feature of their solution is its identity-based nature. The notion of identity-based cryptography dates back to work of Shamir [296]. Shamir’s vision was to do away with public keys and the clumsy certificates for those public keys, and instead build cryptographic schemes and protocols in which entities’ public keys could be derived from their identities (or other identifying information) alone. In place of a Certification Authority (CA), Shamir envisaged a Trusted Authority (TA) who would be responsible for the issuance of private keys and the maintenance of system parameters. Whilst Shamir was able to construct an identity-based signature scheme in [296], and identity-based NIKDS followed from a variety of authors (see [240, p. 587]), the problem of constructing a truly practical and provably secure identity-based encryption scheme remained an open problem until the advent of pairing-based cryptography. As we shall see in Section X.3, the work of

²Note that these problems are defined in Section IX.11.3 for unmodified pairings. We will define the BDH problem for modified pairings below, after which the definition of the DBDH problem should be obvious.

Sakai et al. [284] can be regarded as being pivotal in Boneh and Franklin's solution of this problem.

Sakai et al. make use of a TA who chooses and makes public the *system parameters* of the form $\langle G_1, G_3, \hat{e} \rangle$ (with properties as in Section X.1.2) along with a cryptographic hash function

$$H_1 : \{0, 1\}^* \rightarrow G_1$$

mapping binary strings of arbitrary length onto elements of G_1 . We briefly indicate in Section X.3.1 below how such a hash function can be constructed. The TA also selects but keeps secret a master secret $s \in \mathbb{Z}_r^*$. The TA interacts with A and B , providing each of them with a private key over a confidential and authenticated channel. These private keys depend on s and the individuals' identities: the TA computes as A 's secret the value $S_A = [s]Q_A$, where $Q_A = H_1(\text{ID}_A) \in G_1$ is a publicly computable function of A 's identity. Likewise, the TA gives B the value $S_B = [s]Q_B$, where $Q_B = H_1(\text{ID}_B)$. Because of its role in distributing private keys, the TA is also known as a Private Key Generator (PKG) in these kinds of applications.

Now, with this keying infrastructure in place, consider the equalities:

$$\hat{e}(S_A, Q_B) = \hat{e}([s]Q_A, Q_B) = \hat{e}(Q_A, Q_B)^s = \hat{e}(Q_A, [s]Q_B) = \hat{e}(Q_A, S_B),$$

where we have made use of the bilinearity of $\hat{e}$. On the one hand, A has the secret S_A and can compute $Q_B = H_1(\text{ID}_B)$ using the public hash function H_1 . On the other hand, B can compute Q_A and has the secret S_B . Thus both parties can compute the value $K_{AB} = \hat{e}(Q_A, Q_B)^s$ and, provided they know each others' identifying information, can do so without any interaction at all. A key suitable for use in cryptographic applications can be derived from K_{AB} by the appropriate use of a key derivation function.

A closely related version of this procedure was rediscovered somewhat later by Dupont and Enge [108]. Their scheme works in the unmodified setting and requires that each entity receive two private key components (one in each group G_1 and G_2). The security proof in [108] is easily adapted to the self-pairing setting. The adapted proof models the hash function H_1 as a random oracle and allows the adversary the power to obtain the private keys of arbitrary entities (except, of course, the keys of entities A and B).

The proof shows that the above procedure generates a key $\hat{e}(Q_A, Q_B)$ which cannot be computed by an adversary, provided that the (modified) bilinear Diffie–Hellman problem (BDH problem) is hard. This problem can be stated informally as follows (c.f. the definition in Section IX.11.3):

Bilinear Diffie–Hellman Problem (BDH Problem): given P , $P_1 = [a]P$, $P_2 = [b]P$ and $P_3 = [c]P$ in G_1 with a , b and c selected uniformly at random from $\mathbb{Z}_r^*$, compute

$$\hat{e}(P, P)^{abc}.$$

One implication of the security proof is that the scheme is *collusion resistant*: no coalition of entities excluding A and B can join together and compromise the key K_{AB} . Notice, however, that the TA can generate A and B 's common key for itself – the scheme enjoys (or suffers from, depending on one's point of view and the application in mind) key escrow. For this reason, A and B must trust the TA not to eavesdrop on communications encrypted by this key and not to disclose the key to other parties. In particular, they must trust the TA to adequately check claimants' identities before issuing them with private keys.

For the purpose of comparison, consider the following alternative traditional (i.e., certificate-based) means of realizing a NIKDS. A CA publishes system parameters $\langle E(\mathbb{F}_q), P \rangle$, where P on E is of prime order r . A chooses a private value a , calculates the public value $q_A = [a]P$ and obtains a certificate on ID_A and q_A from a Certification Authority (CA). Entity B does the same with his value b . Now A can compute a common key as follows: A fetches B 's certificate and verifies that it is valid by checking the CA's signature. Now A can combine his secret a with B 's value $[b]P$ to obtain $[ab]P$. This value constitutes the common key. Here, A and B have simply engaged in a non-interactive version of the ECDH protocol. The complexity with this approach comes from the need for A to obtain B 's certificate, verify its correctness and check its revocation status, and vice versa. These checks require the use of a public-key infrastructure (PKI). In contrast, with the identity-based scheme of [284], all A needs is B 's identity string ID_B and the public parameters of the TA.³ This could be B 's e-mail or IP address, or any other string which identifies B uniquely within the context of the system. The trust in public values does not come from certificates but is rather produced implicitly through A 's trust in the TA's private key issuance procedures.

At this point, the reader would be justified in asking: why do A and B simply not use the key K_{AB} as the basis for deriving an encryption key? Moreover, if they do, why does the combination of Sakai et al.'s identity-based NIKDS with this encryption not constitute an identity-based encryption scheme? There are two parts to the answer to this latter question. First of all, the key they agree on is *static*, whereas a dynamic message key would be preferable. Secondly, and more importantly, both A and B must have registered ahead of time and have received their private keys before they can communicate in this way. A true public-key encryption scheme would not require the *encrypting* party to register and obtain such a key.

X.2.2. Three-Party Key Distribution. Around the same time that Sakai et al. proposed their two-party NIKDS, Joux [183] put forward a three-party

³The revocation issue for the identity-based approach also requires careful consideration. We shall return to this topic in Section X.7, where we take a closer look at identity-based systems.

key agreement protocol with the novel feature that only one (broadcast) message per participant is required to achieve key agreement. Thus only one round of communication is needed to establish a shared key. This contrasts sharply with the two rounds that are needed if a naive extension of the (Elliptic Curve) Diffie–Hellman protocol is used. We sketch Joux’s protocol. First of all, it is assumed that the three parties have agreed in advance on system parameters $\langle G_1, G_3, \hat{e}, P \rangle$. Then entity A selects $a \in \mathbb{Z}_r^*$ uniformly at random and broadcasts an ephemeral value $[a]P$ to entities B and C . Entity B (respectively C) selects b (resp. c) in the same way and broadcasts $[b]P$ (resp. $[c]P$) to the other entities. Now by bilinearity we have:

$$\hat{e}([b]P, [c]P)^a = \hat{e}([a]P, [c]P)^b = \hat{e}([a]P, [b]P)^c$$

so that each party, using its private value and the two public values, can calculate the common value

$$K_{ABC} = \hat{e}(P, P)^{abc} \in G_3.$$

This value can be used as keying material to derive session keys. On the other hand, an adversary who only sees the broadcast messages $[a]P$, $[b]P$, $[c]P$ is left with an instance of the BDH problem to solve in order to calculate K_{ABC} . This last statement can be formalized to construct a security proof relating the security of this protocol against *passive* adversaries to the hardness of the (modified) BDH problem. The protocol is vulnerable to an extension of the classic man-in-the-middle attack conducted by an active adversary. We will return to this issue in Section X.6 below.

Note the importance of the fact that $\hat{e}(P, P) \neq 1$ here. Without this condition, K_{ABC} could trivially equal $1 \in G_3$. Joux’s protocol was originally stated in the context of an unmodified pairing and required each participant to broadcast a pair of independent points of the form $[a]P, [a]Q$ in order to avoid degeneracy in the pairing computation. Using modified pairings limits the range of curves for which the protocol can be realized but decreases its bandwidth requirements. This point was first observed by Verheul [335].

X.3. Identity-Based Encryption

As we have discussed above, the construction of a workable and provably secure identity-based encryption (IBE) scheme was, until recently, an open problem dating back to Shamir’s 1984 paper [296]. Two solutions appeared in rapid succession in early 2001 – the pairing-based approach of Boneh and Franklin [38] (appearing in an extended version as [39]) and Cocks’ scheme based on the Quadratic Residuosity problem [83]. It has since become apparent that Cocks’ scheme was discovered some years earlier but remained unpublished until 2001, when the circulation of Boneh and Franklin’s scheme

prompted its disclosure.⁴ We do not discuss Cocks' scheme any further here but recommend that the interested reader consult [83] for the details.

X.3.1. The Basic Scheme of Boneh and Franklin. We first discuss the scheme **BasicIdent** of [39]. This basic IBE scheme is useful as a teaching tool, but is not suited for practical use (because its security guarantees are too weak for most applications). We will study the full scheme **FullIdent** of [39] in Section X.3.3. The IBE scheme **BasicIdent** makes use of essentially the same keying infrastructure as was introduced above in describing the NIKDS of Sakai et al. The TA (or PKG) publishes system parameters $\langle G_1, G_3, \hat{e} \rangle$. In addition, the PKG publishes a generator P for G_1 , together with the point $Q_0 = [s]P$, where, as before, $s \in \mathbb{Z}_r^*$ is a master secret. Note that Q_0 is denoted by P_{pub} in [39]. Descriptions of cryptographic hash functions

$$H_1 : \{0, 1\}^* \rightarrow G_1, \quad H_2 : G_3 \rightarrow \{0, 1\}^n$$

are also made public. Here, n will be the bit-length of plaintext messages. So the complete set of system parameters is

$$\langle G_1, G_3, \hat{e}, P, Q_0, n, H_1, H_2 \rangle.$$

As in the scheme of [284], each entity A must be given a copy of its private key $S_A = [s]Q_A = [s]H_1(\text{ID}_A)$ over a secure channel.

With this set of parameters and keys in place, **BasicIdent** encryption proceeds as follows. To encrypt an n -bit plaintext M for entity A with identity ID_A , entity B computes $Q_A = H_1(\text{ID}_A)$, selects $t \in \mathbb{Z}_r^*$ uniformly at random and computes the ciphertext as:

$$C = \langle [t]P, M \oplus H_2(\hat{e}(Q_A, Q_0)^t) \rangle \in G_1 \times \{0, 1\}^n.$$

To decrypt a received ciphertext $C = \langle U, V \rangle$ in the scheme **BasicIdent**, entity A computes

$$M' = V \oplus H_2(\hat{e}(S_A, U))$$

using its private key $S_A = [s]Q_A$.

To see that encryption and decryption are inverse operations, note that (by bilinearity)

$$\hat{e}(Q_A, Q_0)^t = \hat{e}(Q_A, P)^{st} = \hat{e}([s]Q_A, [t]P) = \hat{e}(S_A, U).$$

On the one hand, the encryption mask $H_2(\hat{e}(Q_A, Q_0)^t)$ that is computed by entity B is the same as that computed by A , namely, $H_2(\hat{e}([s]Q_A, U))$. On the other hand, the computation of the encryption mask by an eavesdropper (informally) requires the computation of $\hat{e}(Q_A, Q_0)^t$ from the values P , Q_A , Q_0 and $U = [t]P$. This task is clearly related to solving the (modified) BDH problem.

⁴Very recently, it has come to our attention that Sakai, Ohgishi and Kasahara proposed an IBE scheme using pairings in May 2000. Their paper was published in Japanese in the proceedings of the 2001 Symposium on Cryptography and Information Security, January 2001; an English version is available from the authors.

Notice that encryption and decryption each require one pairing computation, but that the cost of this can be spread over many encryptions if the encrypting party repeatedly sends messages to the same entity. A small number of other operations is also needed by each entity (dominated by hashing and exponentiation in G_1 and G_3). Ciphertexts are relatively compact: they are equal in size to the plaintext plus the number of bits needed to represent an element of G_1 .

The definition of the hash function H_1 mapping arbitrary strings onto elements of G_1 requires care; a detailed exposition is beyond the scope of this survey. The reader is referred to [39, Sections 4.3 and 5.2] for the details of one approach that works for a particular class of curves and to [42, Section 3.3] for a less elegant method which works for general curves.

X.3.2. Relationship to Earlier Work. It is instructive to examine how this basic identity-based encryption scheme relates to earlier work. There are (at least) two different ways to do so.

Writing $Q_A = [a]P$ for some $a \in \mathbb{Z}_r^*$, we see that the value $\hat{e}(Q_A, Q_0)^t$ appearing in **BasicIdent** is equal to $\hat{e}(P, P)^{ast}$. Thus it is formally equal to the shared value that would be agreed in an instance of Joux's protocol in which the ephemeral values "broadcast" by the entities were $Q_A = [a]P$, $Q_0 = [s]P$ and $U = [t]P$. In the encryption scheme, only U is actually transmitted; the other values are static in the scheme and made available to B through the system parameters and hashing of A 's identity. One can think of $Q_0 = [s]P$ as being the ephemeral value from a "dummy" entity here. Entity A gets the value U from B and is given the ability to compute $\hat{e}(P, P)^{ast}$ when the PKG gives it the value $[s]Q_A = [sa]P$. Thus Boneh and Franklin's IBE scheme can be regarded as a rather strange instance of Joux's protocol.

Perhaps a more profitable way to understand the scheme is to compare it to ElGamal encryption. In a variant of textbook ElGamal, an entity A has a private key $x_A \in \mathbb{Z}_r^*$ and a public key $y_A = g^{x_A}$. To encrypt a message for A , entity B selects $t \in \mathbb{Z}_r^*$ uniformly at random and computes the ciphertext as:

$$C = \langle g^t, M \oplus H_2(y_A^t) \rangle$$

while to decrypt $C = \langle U, V \rangle$, entity A computes

$$M' = V \oplus H_2(U^{x_A}).$$

Thus one can regard the basic IBE scheme of Boneh and Franklin as being an adaptation of ElGamal encryption in which $\hat{e}(Q_A, Q_0)$, computed from system parameters and A 's identity, replaces the public key y_A .

We have already noted the similarities in keying infrastructures used by Boneh and Franklin's IBE scheme and in the NIKDS of Sakai *et al.* [284]. The above discussion shows a relationship between Boneh and Franklin's IBE scheme and Joux's protocol [183]. However, it would be wrong to leave the impression that Boneh and Franklin's scheme is just a simple development

of ideas in these earlier papers. Prior to Boneh and Franklin's work, Joux's protocol was merely an interesting curiosity and the work of [284] almost unknown to the wider community. It was Boneh and Franklin's work that quickly led to a wider realization that pairings could be a very useful constructive cryptographic tool and the spate of research that followed.

X.3.3. Security of Identity-Based Encryption. Boneh and Franklin provide in [39] a variant of **BasicIdent** named **FullIdent** which offers stronger security guarantees. In particular, the security of **FullIdent** can be related to the hardness of the BDH problem in a model that naturally extends the widely-accepted IND-CCA2 model for public-key encryption (see Definition III.4) to the identity-based setting. We present the scheme **FullIdent** below, outline the security model introduced in [39] and then discuss the security of **FullIdent** in this model.

In general, an IBE scheme can be defined by four algorithms, with functions as suggested by their names: **Setup**, (**Private Key**) **Extract**, **Encrypt** and **Decrypt**. For the scheme **FullIdent**, these operate as follows:

Setup: This algorithm takes as input a security parameter ℓ and outputs the system parameters:

$$\text{params} = \langle G_1, G_3, \hat{e}, n, P, Q_0, H_1, H_2, H_3, H_4 \rangle.$$

Here G_1 , G_3 and $\hat{e}$ are the usual objects⁵, n is the bit-length of plaintexts, P generates G_1 and $Q_0 = [s]P$, where s is the scheme's master secret. Hash functions H_1 and H_2 are as above, while $H_3 : \{0, 1\}^{2n} \rightarrow \mathbb{Z}_r^*$ and $H_4 : \{0, 1\}^n \rightarrow \{0, 1\}^n$ are additional hash functions. In principle, all of these parameters may depend on ℓ .

Extract: This algorithm takes as input an identity string ID and returns the corresponding private key $[s]H_1(\text{ID})$.

Encrypt: To encrypt the plaintext $M \in \{0, 1\}^n$ for entity A with identity ID_A , perform the following steps:

1. Compute $Q_A = H_1(\text{ID}_A) \in G_1$.
2. Choose a random $\sigma \in \{0, 1\}^n$.
3. Set $t = H_3(\sigma, M)$.
4. Compute and output the ciphertext:

$$C = \langle [t]P, \sigma \oplus H_2(\hat{e}(Q_A, Q_0)^t), M \oplus H_4(\sigma) \rangle \in G_1 \times \{0, 1\}^{2n}.$$

Decrypt: Suppose $C = \langle U, V, W \rangle \in G_1 \times \{0, 1\}^{2n}$ is a ciphertext encrypted for A . To decrypt C using the private key $[s]Q_A$:

1. Compute $\sigma' := V \oplus H_2(\hat{e}([s]Q_A, U))$.
2. Compute $M' := W \oplus H_4(\sigma')$.

⁵Boneh and Franklin make use of a subsidiary instance generating algorithm $\mathcal{IG}$ to produce the parameters $\langle G_1, G_3, \hat{e} \rangle$ (possibly probabilistically) from input ℓ , the security parameter.

3. Set $t' = H_3(\sigma', M')$ and test if $U = [t']P$. If not, reject the ciphertext.
4. Otherwise, output M' as the decryption of C .

The reader should compare **FullIdent** with the basic scheme above. When C is a valid encryption of M , it is quite easy to see that decrypting C will result in an output $M' = M$. The value $H_2(e(Q_A, Q_0)^t)$ is still used as an encryption mask, but now it encrypts a string σ rather than the plaintext itself. The string σ is subsequently used to form an encryption key $H_4(\sigma)$ to mask the plaintext. The encryption process also now derives t by hashing rather than by random choice; this provides the decryption algorithm with a checking facility to reject ciphertexts that are not of the correct form.

In fact, the scheme **FullIdent** is obtained from the basic scheme of the previous section by applying the Fujisaki–Okamoto hybridization technique [128]. It is this technique that ensures **FullIdent** meets the strong security definition in the model developed by Boneh and Franklin in [39]. In that model, an adversary $\mathcal{A}$ plays against a challenger $\mathcal{C}$ in the following game:

IND-ID-CCA Security Game: The game runs in five steps:

Setup: $\mathcal{C}$ runs algorithm **Setup** on input some value ℓ , gives $\mathcal{A}$ the system parameters **params** and keeps the master secret s to itself.

Phase 1: $\mathcal{A}$ issues a series of queries, each of which is either an **Extract** query on an identity, in which case $\mathcal{C}$ responds with the appropriate private key, or a **Decrypt** query on an identity/ciphertext combination, in which case $\mathcal{C}$ responds with an appropriate plaintext (or possibly a fail message).

Challenge: Once $\mathcal{A}$ decides to end Phase 1, it selects two plaintexts M_0, M_1 and an identity ID_{ch} on which it wishes to be challenged. We insist that ID_{ch} not be the subject of an earlier **Extract** query. Challenger $\mathcal{C}$ then chooses b at random from $\{0, 1\}$ and runs algorithm **Encrypt** on M_b and ID_{ch} to obtain the challenge ciphertext C^* ; $\mathcal{C}$ then gives C^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ issues another series of queries as in Phase 1, with the restriction that no **Extract** query be on ID_{ch} and that no **Decrypt** query be on the combination $\langle \text{ID}_{\text{ch}}, C^* \rangle$. $\mathcal{C}$ responds to these as before.

Guess: Finally, $\mathcal{A}$ outputs a guess b' and wins the game if $b' = b$.

Adversary $\mathcal{A}$'s advantage is defined to be $\text{Adv}(\mathcal{A}) := 2|\Pr[b' = b] - \frac{1}{2}|$, where the probability is measured over any random bits used by $\mathcal{C}$ (for example, in the **Setup** algorithm) and $\mathcal{A}$ (for example, in choosing ciphertexts and identities to attack). An IBE scheme is said to be semantically secure against an adaptive chosen ciphertext attack (IND-ID-CCA secure) if no polynomially bounded adversary $\mathcal{A}$ has a non-negligible advantage in the above game. Here, non-negligibility is defined in terms of the security parameter ℓ used

in the **Setup** algorithm.⁶ This model and definition of security extends the by-now-standard IND-CCA2 notion of security for public-key encryption: it allows the adversary to access private keys of arbitrary entities (except the challenge identity, of course) as well as giving the adversary access to a decryption oracle. It also allows the adversary to choose the public key on which it is to be challenged and automatically captures attacks involving colluding entities.

It is proved in [39] that the scheme **FullIdent** is IND-ID-CCA secure in the Random Oracle model, provided that there is no polynomially bounded algorithm having a non-negligible advantage in solving the BDH problem. Here, parameters $\langle G_1, G_2, \hat{e} \rangle$ for the BDH problem are assumed to be generated with the same distribution as by the **Setup** algorithm of **FullIdent**.

The proof of security for **FullIdent** proceeds in several stages. First it is shown, via a fairly standard simulation argument, that an adversary who can break **FullIdent** (in the sense of winning the IND-ID-CCA security game) can be used to produce an adversary that breaks a related standard public-key encryption scheme in an IND-CCA2 game. Then results of [128] are invoked to relate the IND-CCA2 security of the public-key scheme to the security of a simpler public-key encryption scheme **BasicPub**, but in a much weaker attack model (one without decryption queries). Finally, it can be shown directly that an adversary breaking **BasicPub** can be used to construct an algorithm to solve instances of the BDH problem. For details of these steps, see [39, Lemma 4.3, Lemma 4.6 and Theorem 4.5].⁷ The security analysis in [39] depends in a crucial way on the replacement of hash functions H_1, H_2, H_3 and H_4 by random oracles. At the time of writing, it is still an open problem to produce an IBE scheme that is provably secure in Boneh and Franklin's security model, but without modelling any hash functions as random oracles. The composition of a sequence of security reductions also yields a fairly loose relationship between the security of **FullIdent** and the hardness of the BDH problem. Tightening this relationship seems to be a difficult challenge.

This concludes our description of the identity-based encryption scheme of Boneh and Franklin [39]. The paper [39] contains much else of interest besides, and we recommend it be read in detail by every reader who has more than a passing interest in the subject.

X.3.4. Further Encryption Schemes. In [335], Verheul showed how pairings can be used to build a scheme supporting both non-repudiable signatures and escrowable public-key encryption using only a single public key.

⁶A function f of ℓ is said to be *negligible* if, for any polynomial $p(\ell)$, there exists ℓ_0 such that, for all $\ell > \ell_0$, $f(\ell) < 1/p(\ell)$. Naturally, a function is said to be *non-negligible* if it is not negligible.

⁷But note that the proof of Lemma 4.6 in [39] requires a small repair: when $\text{coin}_i = 1$, the values b_i should be set to equal 1, so that the ciphertexts C'_i do not always fail the consistency check in the decryption algorithm of **BasicPub**^{hy}.

The main idea of Verheul's scheme is as follows. As usual, we have system parameters $\langle G_1, G_3, \hat{e} \rangle$ with G_1 of prime order r generated by point P . An entity A chooses as its private signing key $x_A \in \mathbb{Z}_r^*$; the corresponding public key used for both encryption and signatures is $y_A = \hat{e}(P, P)^{x_A} \in G_3$. A CA then issues A with a certificate on the value y_A (the scheme is not identity-based). Any discrete logarithm based digital signature algorithm employing the values $g = \hat{e}(P, P)$, x_A and $y_A = g^{x_A}$ can be used. To encrypt a message $M \in \{0, 1\}^n$ for A , the sender generates a random $t \in \mathbb{Z}_r^*$ and computes the ciphertext:

$$C = \langle [t]P, M \oplus H_2((y_A)^t) \rangle.$$

Here, as before, $H_2 : G_3 \rightarrow \{0, 1\}^n$ is a cryptographic hash function. To decrypt $C = \langle U, V \rangle$, entity A computes

$$M' = V \oplus H_2(\hat{e}(P, U)^{x_A}).$$

Notice the similarity of this encryption scheme to that in Section X.3.2. The escrow service is supported as follows. Ahead of time, A sends to the escrow agent the value $Y_A = [x_A]P$. The escrow agent can then calculate the value $\hat{e}(P, U)^{x_A}$ for itself using its knowledge of Y_A and bilinearity:

$$\hat{e}(Y_A, U) = \hat{e}([x_A]P, U) = \hat{e}(P, U)^{x_A}.$$

Note that A does not give up its private signing key x_A to the escrow agent. Thus A 's signatures remain non-repudiable. Verheul's scheme currently lacks a formal security proof. Such a proof would show that the same public key can safely be used for both signature and encryption.

Verheul's scheme may be described as providing a non-global escrow: entity A must choose to send the value Y_A to the escrow agent in order that the agent may recover plaintexts. Boneh and Franklin in [39, Section 7] gave yet another variant of pairing-based ElGamal encryption that provides escrow yet does not require interaction between escrow agent and users. For this reason, they described their scheme as providing global escrow. Their scheme works as follows. The system parameters, chosen by the escrow agent, are $\langle G_1, G_3, \hat{e}, P, Q_0, n, H_2 \rangle$. These are all defined as for the basic IBE scheme in Section X.3.1. In particular, $Q_0 = [s]P$, where s is a master secret. An entity A 's key-pair is of the form $\langle x_A, Y_A = [x_A]P \rangle$. Thus A 's public key is identical to the escrowed key in Verheul's scheme, and A 's private key is the same in the two schemes. Now to encrypt $M \in \{0, 1\}^n$ for A , the sender generates a random $t \in \mathbb{Z}_r^*$ and computes the ciphertext:

$$C = \langle [t]P, M \oplus H_2(\hat{e}(Y_A, Q_0)^t) \rangle.$$

To decrypt $C = \langle U, V \rangle$, entity A computes

$$M' = V \oplus H_2(\hat{e}([x_A]Q_0, U))$$

while the escrow agent computes

$$M' = V \oplus H_2(\hat{e}([s]Y_A, U)).$$

It is straightforward to see that (by bilinearity) both decryption algorithms produce the plaintext M . It is claimed in [39] that the security of this scheme rests on the hardness of the BDH problem. To see informally why this is so, note that to decrypt an adversary must compute the value $\hat{e}(P, P)^{stx_A}$ given the values $Q_0 = [s]P$, $U = [t]P$ and $Y_A = [x_A]P$.

Lynn [228] has shown how to combine ideas from the IBE scheme of [39] and the NIKDS of [284] to produce an *authenticated identity-based encryption scheme*. In this scheme, a recipient A can check which entity sent any particular ciphertext. Simplifying slightly, this ability is provided by using the NIKDS key $\hat{e}(Q_A, Q_B)^s$ in place of the value $\hat{e}(Q_A, Q_0)^r$ in the Boneh–Franklin IBE scheme. This approach cannot yield a non-repudiation service, since A itself could have prepared any authenticated ciphertext purported to be from B .

We will report on the hierarchical identity-based encryption scheme of Gentry and Silverberg [147] and related work in Section X.5.

X.4. Signature Schemes

In this section we outline how pairings have been used to build signature schemes of various kinds. Our coverage includes identity-based signature and signcryption schemes, standard (i.e., not identity-based) signature schemes and a variety of special-purpose signature schemes.

X.4.1. Identity-Based Signature Schemes. Not long after the appearance of Boneh and Franklin’s IBE scheme, a rash of identity-based signature (IBS) schemes appeared [62, 163, 164, 272]. Sakai et al.’s paper [284] also contains an IBS; another IBS scheme appears in [350]. Since IBS schemes have been known since Shamir’s original work on identity-based cryptography in [296], the main reason to be interested in these new schemes is that they can make use of the same keying infrastructure as the IBE scheme of [39]. Being identity-based, and hence having built in escrow of private keys, none of the schemes can offer a true non-repudiation service. The schemes offer a variety of trade-offs in terms of their computational requirements on signer and verifier, and signature sizes. The scheme of [62] enjoys a security proof in a model that extends the standard adaptive chosen message attack model for (normal) signature schemes of [150] to the identity-based setting. The proof is in the random oracle model and relates the scheme’s security to the hardness of the computational Diffie–Hellman problem (CDH problem) in G_1 using the Forking Lemma methodology [276]. The first IBS scheme of [163] also has a security proof; the second scheme in [163] was broken in [75].

To give a flavour of how these various IBS schemes operate, we present a version of the scheme of Cha and Cheon [62] here. An IBS scheme is defined by four algorithms: **Setup**, **Extract**, **Sign** and **Verify**. For the scheme of [62], these operate as follows:

Setup: This algorithm takes as input a security parameter ℓ and outputs the system parameters:

$$\text{params} = \langle G_1, G_3, \hat{e}, P, Q_0, H_1, H_2 \rangle.$$

Here G_1 , G_3 , $\hat{e}$, P and $Q_0 = [s]P$ are as usual; s is the scheme's master secret. The hash function $H_1 : \{0, 1\}^* \rightarrow G_1$ is as in Boneh and Franklin's IBE scheme, while $H_2 : \{0, 1\}^* \times G_1 \rightarrow \mathbb{Z}_r$ is a second hash function.

Extract: This algorithm takes as input an identity ID and returns the corresponding private key $S_{\text{ID}} = [s]H_1(\text{ID})$. Notice that this key is identical to the private key in the IBE scheme of Boneh and Franklin [39].⁸

Sign: To sign a message $M \in \{0, 1\}^*$, entity A with identity ID_A and private key $S_A = [s]H_1(\text{ID}_A)$ chooses a random $t \in \mathbb{Z}_r$ and outputs a signature $\sigma = \langle U, V \rangle$ where $U = [t]H_1(\text{ID}_A)$, $h = H_2(M, U)$ and $V = [t + h]S_A$.

Verify: To verify a signature $\sigma = \langle U, V \rangle$ on a message M for identity ID_A , an entity simply checks whether the equation

$$\hat{e}(Q_0, U + hQ_A) = \hat{e}(P, V)$$

holds.

It is a simple exercise to show that the above IBS scheme is sound (signatures created using **Sign** will verify correctly using **Verify**).

The IBS scheme of [62] was originally presented in the context of any gap Diffie–Hellman group. Informally speaking, these are groups in which the CDH problem is hard but the DDH problem is easy, a notion first formalized in [263] and further explored in [186]. The signature generation algorithm uses the private key D_A to create Diffie–Hellman tuples while the signature verification algorithm amounts to deciding whether $\langle P, Q_0, U + hQ_A, V \rangle$ is a valid Diffie–Hellman tuple. Since all the realizations of such gap groups currently known use pairings on elliptic curves, we have preferred a presentation using pairings.

X.4.2. Short Signatures. In [42, 43], Boneh, Lynn and Shacham used pairings to construct a (normal) signature scheme in which the signatures are rather short: for example, one version of their scheme has signatures that are approximately 170 bits in length whilst offering security comparable to that of 320-bit DSA signatures.

A simplified version of this BLS scheme can be described using modified pairings though (for reasons which will be discussed below) this does not lead to the preferred instantiation. This is essentially the approach taken in [42]. We will begin with this approach for ease of presentation.

⁸It is generally good cryptographic practice to use different keys for different functions. If this is required here, then a separate master secret could be used for the IBS scheme, or the identity string ID could be replaced by the string $\text{ID}||\text{“Sig”}$, where $||$ denotes a concatenation of strings.

As usual, we work with system parameters $\langle G_1, G_3, \hat{e} \rangle$ and assume P of prime order r generates G_1 . We also need a hash function $H : \{0, 1\}^* \rightarrow G_1$. A user's private key is a value x selected at random from $\mathbb{Z}_r$, and the matching public key is $[x]P \in G_1$. The signature on a message $M \in \{0, 1\}^*$ is simply $\sigma = [x]H(M) \in G_1$. To verify a purported signature σ on message M , the verifier checks that the 4-tuple:

$$\langle P, [x]P, H(M), \sigma \rangle$$

is a Diffie–Hellman tuple. This can be done by checking that the equation:

$$\hat{e}(\sigma, P) = \hat{e}(H(M), [x]P)$$

holds.

As with the IBS scheme of [62], this signature scheme exploits the fact that the signer can create Diffie–Hellman tuples in G_1 using knowledge of the private key x while the verifier can check signatures using the fact that the DDH problem is easy in G_1 , thanks to the presence of the pairing $\hat{e}$. The scheme is very closely related to the undeniable signature scheme of Chaum and van Antwerpen [66, 67]. That scheme has an identical signing procedure (except for a change of notation), but the confirmation (or denial of a signature) is via a zero-knowledge protocol in which the signer proves (or disproves) that the tuple is a Diffie–Hellman tuple. One can view the scheme of [42] as being the result of replacing the confirmation and denial protocols by a pairing computation. This makes the signatures verifiable without the aid of the signer, thus converting the undeniable signature scheme into a standard one. Of course, the BLS construction works more generally in the setting of gap Diffie–Hellman groups; the observation that signature schemes could be constructed from gap problems was made in [263, Section 4.1], though without a specific (standard) scheme being presented. The scheme of [42] can also be viewed in another way. As is noted in [39], Naor has pointed out that any IBE scheme can be used to construct a signature scheme as follows: the private signing key is the master key for the IBE scheme, the public verification key is the set of public parameters of the IBE scheme, and the signature on a message M is simply the private key for “identity” M in the IBE scheme. To verify a signature, the verifier can encrypt a random string and check that the signature (viewed as a decryption key) properly decrypts the result. In the special case of the IBE scheme of Boneh and Franklin, the signature for message M would be the IBE private key $[s]H_1(M)$. This is simply a BLS signature on M . The BLS scheme replaces the trial encryption/decryption with a more efficient procedure, but it is otherwise the signature scheme that can be derived from the Boneh–Franklin IBE scheme using Naor’s construction.

It is not difficult to show that the BLS signature scheme is secure (in the usual chosen message attack model of [150] and regarding H as a random oracle) provided the CDH problem is hard in G_1 .

A signature in this scheme consists of a single element of G_1 (as does the public key). Thus short signatures will result whenever G_1 can be arranged to have a compact representation. Using point compression, elements of G_1 can be represented using roughly $\lceil \log_2 q \rceil$ bits if G_1 is a subgroup of $E(\mathbb{F}_q)$.⁹ So in order to obtain signatures that are as short as possible, it is desirable to make q as small as possible whilst keeping the ECDHP in G_1 (a subgroup of $E(\mathbb{F}_q)$) hard enough to make the scheme secure. However, one must bear in mind that, because of the presence of the pairing $\hat{e}$, the ECDLP in $E(\mathbb{F}_q)$ can be translated via the MOV reduction into the DLP in $\mathbb{F}_{q^k}$, where k is the embedding degree of $E(\mathbb{F}_q)$. Thus the security of the scheme not only rests on the difficulty of solving the ECDHP in $E(\mathbb{F}_q)$, but also on the hardness of the DLP in $\mathbb{F}_{q^k}$.

At first sight, it seems that Table IX.1 gives a pair of characteristic 3 supersingular curves E_1, E_2 which are fit for purpose.¹⁰ When ℓ is odd, the curves have embedding degree 6, so the MOV reduction translates the ECDLP on $E_i(\mathbb{F}_{3^\ell})$ into the DLP in $\mathbb{F}_{3^{6\ell}}$, a relatively large finite field. Thus it should be possible to select a moderate sized ℓ and obtain short, secure signatures. For example, according to [43, Table 2], taking $\ell = 121$, one can obtain a signature size of 192 bits for a group G_1 of size about 2^{155} while the MOV reduction yields a DLP in $\mathbb{F}_{3^{726}}$, a field of size roughly 2^{1151} . This set of parameters would therefore appear to offer about 80 bits of security.¹¹

However, as is pointed out in [43], Coppersmith's discrete logarithm algorithm [86], although specifically designed for fields of characteristic 2, also applies to fields of small characteristic and is more efficient than general purpose discrete logarithm algorithms. The function field sieve as developed in [2, 3, 185] is also applicable and has better asymptotic performance than Coppersmith's algorithm for fields of characteristic 3. But it is currently unclear by how much these algorithms reduce the security offered by BLS signatures for particular curves defined over fields of characteristic 3. For example, it may well be that the algorithm reduces the security level below the supposed 80 bits for the parameters in the paragraph above. The conclusion of [43] is that in order to obtain security similar to that offered by DSA, curves $E_i(\mathbb{F}_{3^\ell})$, where $3^{6\ell}$ is much greater than 1024 bits in size, are needed. Similar security considerations apply when using the same curves in other cryptographic applications. In the current context, this results in much longer signatures, running counter to the whole rationale for the BLS scheme. The problem of constructing signatures that are simultaneously short and secure should provide motivation for a detailed study of the performance of the

⁹A modified verification equation is then needed to handle the fact that two elements of G_1 are represented by each $x \in \mathbb{F}_q$.

¹⁰These curves are named E^+, E^- in [42].

¹¹This choice of parameters was not present in the original version [42] because of the threat of Weil descent attacks; according to [43], the work of Diem in [104] shows Weil descent to be ineffective for $\ell = 121$.

function field sieve in characteristic 3. Some estimates for the size of factor bases arising in the function field sieve for fields of small characteristic can be found in [154].

In [43], Boneh, Lynn and Shacham explain how ordinary (i.e., non-supersingular) curves and unmodified pairings can be used to remedy the situation. Assume now we have a triple of groups G_1 , G_2 , G_3 and a pairing $e : G_1 \times G_2 \rightarrow G_3$. For $i = 1, 2$, let P_i of prime order r generate G_i . A user's private key is still a value $x \in \mathbb{Z}_r$, but now the matching public key is $[x]P_2 \in G_2$. The signature on a message $M \in \{0, 1\}^*$ is still $\sigma = [x]H(M) \in G_1$. To verify a purported signature σ on message M , the verifier now checks that

$$\langle P_2, [x]P_2, H(M), \sigma \rangle$$

is a valid co-Diffie–Hellman tuple, that is, a tuple in which the second pair of elements (in G_1) are related by the same multiple as the first pair (in G_2). This can be done using the pairing e by checking that the equation:

$$e(\sigma, P_2) = e(H(M), [x]P_2)$$

holds. The security of this scheme rests on the hardness of the co-CDH problem, a variant of the CDH problem appropriate to the situation where two groups G_1 and G_2 are in play. The security proof has an interesting twist, in that the existence of an efficiently computable isomorphism $\psi : G_2 \rightarrow G_1$ is required to make the proof work.

Boneh, Lynn and Shacham [42] show how groups and pairings suitable for use with this scheme can be obtained from MNT curves (see Section IX.15.1) and how ψ can be constructed using the trace map. They report an example curve $E(\mathbb{F}_q)$ where q is a 168-bit prime and where the embedding degree is 6. The curve has an order that is divisible by a 166-bit prime r ; using appropriate subgroups of $E(\mathbb{F}_q)$ and $E(\mathbb{F}_{q^6})$ for G_1 and G_2 , one can obtain a scheme with 168-bit signatures where the best currently known algorithm for the co-CDH problem requires either a generic discrete logarithm algorithm using around 2^{83} computational steps or taking a discrete logarithm in a 1008-bit field of large characteristic (where Coppersmith's algorithm and the function field sieve are ineffective). Unfortunately, the public key, being a point on $E(\mathbb{F}_{q^6})$, is no longer short, an issue that may limit the wider applicability of this scheme.

The above discussion gives a clear example where unmodified pairings should be used in preference to modified pairings for reasons of efficiency and security.

X.4.3. Further Signature Schemes. We provide brief references to a selection of the other relevant literature.

Libert and Quisquater developed an identity-based undeniable signature scheme in [222]. Pairings were used to construct a variety of proxy signature schemes by Zhang et al. in [359]. Identity-based blind signatures and ring signatures were considered by Zhang and Kim in [354, 355], but the

schemes presented lack a full security analysis. Herranz and Sáez [162] used the Forking Lemma methodology to build provably secure identity-based ring signatures from pairings.

Thanks mainly to their simple algebraic structure, BLS signatures have been productively exploited by a number of authors. Boldyreva [32] showed how to adapt the scheme of [42] to produce provably secure threshold signatures, multisignatures and blind signatures. The blinding capability of BLS signatures was also noted by Verheul in [337]. In the same paper, Verheul also considered the use of pairings to construct self-blindable credential certificates. Steinfeld et al. [318] extended the BLS signature scheme to obtain a new primitive, universal designated-verifier signatures. Boneh et al. [40] also used BLS signatures as a basis to produce an aggregate signature scheme (in which multiple signatures can be combined to form a single, short, verifiable signature), a verifiably encrypted signature scheme (with applications to fair exchange and optimistic contract signing), and a ring signature scheme. In turn, Boldyreva et al. [33] used the aggregate signature scheme of [40] to construct efficient proxy signature schemes. See also [166] for an attack on and repair of the verifiably encrypted signature scheme of [40] and [89] for a result relating the complexity assumption that was used to establish security for the aggregate signature scheme in [40] to the CDH problem.

Recently, Libert and Quisquater [223] modified the BLS signature scheme to produce a particularly efficient *signcryption* scheme, that is, a scheme in which the signature and encryption are combined into a single “monolithic” operation. An alternative scheme of Malone-Lee [231] has a security proof in a multi-user model and offers ciphertexts that are even shorter than in the scheme of [223]. Malone-Lee’s scheme is not based on BLS signatures but does use pairings as a tool in the security proofs.

Zhang et al. [358] modified the BLS signature scheme to obtain a more efficient signature scheme that does not require the use of a special hash function (i.e., one that outputs elements of G_1). The scheme is provably secure in the random oracle model, but its security is based on the hardness of the non-standard k -weak CDH problem that was introduced in [250]. Zhang et al. [357] adapted the scheme of [358] to obtain a verifiably encrypted signature scheme, also based on pairings, but more efficient than the scheme of [40].

Boneh, Mironov and Shoup [44] used pairings to construct a tree-based signature scheme whose security can be proved in the standard model (i.e., without the use of random oracles), based on the hardness of the CDH problem. A much more efficient scheme, also secure in the standard model, was presented in [35]. Here, the security relies on the hardness of another non-standard problem, the Strong Diffie–Hellman problem. This problem is related to the k -weak CDH problem of [250].

X.4.4. Identity-Based Signcryption. A number of authors have considered combining signature and encryption functions in a single identity-based scheme. The first attempt appears to be that of Malone-Lee [230], who provided an identity-based signcryption scheme. Unfortunately, the computational costs of the signcryption and matching un-signcryption operations in [230] are not much less than the sum of the costs of the encryption/decryption and signature/verification algorithms of [39] and [62] (say). On the other hand, the scheme's ciphertexts are a little shorter than they would be in the case of a simple "sign then encrypt" scheme. In contrast to the scheme of Lynn [228], Malone-Lee's scheme offers non-repudiation: an entity A can present a message and ciphertext to a judge who can then verify that they originated from another entity B . However, as is pointed out in [221], this property means that Malone-Lee's scheme cannot be semantically secure.¹² An identity-based signcryption scheme which does not suffer from this weakness was presented by Libert and Quisquater in [221]. The scheme uses pairings, is roughly as efficient as the scheme of [230] and has security that depends on the hardness of the decision bilinear Diffie–Hellman problem (defined in Section IX.11.3 for unmodified pairings). This scheme also allows non-repudiation, but the origin of ciphertexts can be verified by third parties without knowledge of the underlying plaintext. This last feature may be a positive or negative one depending on the intended application.

A two-layer approach to combining identity-based signature and encryption was taken by Boyen in [48]. The resulting mechanism, called an IBSE scheme, has comparable efficiency but stronger security guarantees than the earlier work of [221, 230]. As well as providing the usual properties of confidentiality and non-repudiation, the pairing-based scheme of Boyen in [48] offers ciphertext unlinkability (allowing the sender to disavow creating a ciphertext), ciphertext authentication (allowing the recipient to be convinced that the ciphertext and signed message it contains were prepared by the same entity) and ciphertext anonymity (making the identification of legitimate sender and recipient impossible for any entity not in possession of the recipient's decryption key, in contrast to the scheme of [221]). These properties are not available from single-layer signcryption schemes and a major contribution of [48] is to identify and formalize these properties. The security of Boyen's IBSE scheme depends on the hardness of the BDH problem. An examination of the scheme shows that it builds on the NIKDS of Sakai et al. [284], with the key $\hat{e}(Q_A, Q_B)^s$ once again being at the heart of the matter. Chen and Malone-Lee [72] have recently proposed an identity-based signcryption scheme that is secure in the model of [48] but more efficient than Boyen's IBSE scheme.

¹²The adversary, when presented with a challenge ciphertext C^* which encrypts one of M_0, M_1 , can simply attempt to verify both pairs M_0, C^* and M_1, C^* ; a correct verification reveals which plaintext M_b was encrypted.

X.5. Hierarchical Identity-Based Cryptography and Related Topics

Identity-based cryptography as we have described it so far in this chapter involves a single trusted authority, the PKG, who carries out all the work of registering users and distributing private keys. Public-key infrastructures (PKIs) supporting “classical” public-key cryptography allow many levels of trusted authority through the use of certificates and certificate chains. A hierarchy of CAs topped by a root CA can spread the workload and simplify the deployment of systems relying on public-key cryptography. The first attempt to mimic the traditional PKI hierarchy in the identity-based setting was due to Horowitz and Lynn [172]. Their scheme is restricted to two levels of hierarchy and has limited collusion resistance. A more successful attempt was made soon after by Gentry and Silverberg [147]. Their solution, which extends the IBE scheme of Boneh and Franklin in a very natural way, has led other researchers to develop further interesting cryptographic schemes. In this section we outline the contribution of Gentry and Silverberg in [147] and then give a brief overview of the subsequent research.

X.5.1. The Basic Scheme of Gentry and Silverberg. The basic hierarchical identity-based encryption (HIBE¹³) scheme of [147] associates each entity with a level in the hierarchy, with the root authority being at level 0. An entity at level t is defined by its tuple of identities $\langle \text{ID}_1, \text{ID}_2, \dots, \text{ID}_t \rangle$. This entity has as superior entities the root authority (or root PKG) together with the $t - 1$ entities whose identities are $\langle \text{ID}_1, \text{ID}_2, \dots, \text{ID}_i \rangle$, $1 \leq i < t$. An entity at level t will have a secret $s_t \in \mathbb{Z}_r^*$, just like the PKG in the Boneh–Franklin IBE scheme. As we describe below, this secret will be used by an entity at level t to produce private keys for its children at level $t + 1$.

The scheme BasicHIBE¹⁴ is defined by five algorithms:

Root Setup, Lower-Level Setup,
(Private Key) Extract, Encrypt and Decrypt.

These operate as follows:

Root Setup: To set up the root authority at level 0, this algorithm takes as input a security parameter ℓ and outputs the system parameters:

$$\text{params} = \langle G_1, G_3, \hat{e}, n, P_0, Q_0, H_1, H_2 \rangle.$$

Here G_1 , G_3 , $\hat{e}$, n (the bit-length of plaintexts) and hash functions H_1 and H_2 are just as in the Boneh–Franklin scheme. We write P_0 for an arbitrary

¹³This is a perhaps more natural acronym than HIDE as used by Gentry and Silverberg, albeit one that does not have the same neat connotation of secrecy. It also enables us to use the acronym HIBS for the matching concept of a hierarchical identity-based signature scheme. It can be no bad thing to mention at least one Scottish football team in this chapter.

¹⁴BasicHIDE in [147].

generator of G_1 and $Q_0 = [s_0]P_0$, where $s_0 \in \mathbb{Z}_r^*$ is the root authority's secret value. Apart from these minor changes of notation, this procedure is identical to the **Setup** procedure of the scheme **BasicIdent** in [39].

Lower-Level Setup: An entity at level t in the hierarchy is initialized simply by selecting for itself a secret value $s_t \in \mathbb{Z}_r^*$.

Extract: Consider a level t entity $\mathcal{E}_t$ with identity tuple $\langle \text{ID}_1, \text{ID}_2, \dots, \text{ID}_t \rangle$. This entity's parent (having identity $\langle \text{ID}_1, \text{ID}_2, \dots, \text{ID}_{t-1} \rangle$) performs the following steps:

1. Compute $P_t = H_1(\text{ID}_1, \text{ID}_2, \dots, \text{ID}_t) \in G_1$.
2. Set $S_t = S_{t-1} + s_t P_t \in G_1$ and give the private key S_t to entity $\mathcal{E}_t$ over a secure channel. (When $t = 1$, we set $S_0 = 1_{G_1}$.)
3. Give $\mathcal{E}_t$ the values $Q_i = s_i P_0$, $1 \leq i < t$.

Notice that, by induction, we have $S_t = \sum_{i=1}^t s_{i-1} P_i$.

Encrypt: To encrypt plaintext $M \in \{0, 1\}^n$ for an entity with identity tuple $\langle \text{ID}_1, \text{ID}_2, \dots, \text{ID}_t \rangle$, perform the following steps:

1. Compute $P_i = H_1(\text{ID}_1, \text{ID}_2, \dots, \text{ID}_i) \in G_1$ for $1 \leq i \leq t$.
2. Choose a random $t \in \mathbb{Z}_r^*$.
3. Compute and output the ciphertext:

$$C = \langle [t]P_0, [t]P_2, \dots, [t]P_t, M \oplus H_2(\hat{e}(P_1, Q_0)^t) \rangle \in G_1^t \times \{0, 1\}^n.$$

Notice that, in order to encrypt a message for an entity, the sender needs only know the parameters of the root PKG along with the identity tuple of the intended recipient, and not any parameters associated with intermediate entities. Note too that the omission of the value $[t]P_1$ from the ciphertext is deliberate (if it were included, then an eavesdropper could decrypt C by calculating the mask $H_2(\hat{e}([t]P_1, Q_0))$).

Decrypt: Suppose $C = \langle U_0, U_2, \dots, U_t, V \rangle \in G_1^t \times \{0, 1\}^n$ is a ciphertext encrypted for an entity $\langle \text{ID}_1, \text{ID}_2, \dots, \text{ID}_t \rangle$. To decrypt C using the private key S_t , the recipient computes

$$M' = V \oplus H_2 \left(\hat{e}(S_t, U_0) \cdot \prod_{i=2}^t \hat{e}(Q_{i-1}, U_i)^{-1} \right).$$

To see that decryption works properly, consider the following chain of equalities, established using the bilinearity of $\hat{e}$:

$$\begin{aligned}
 \hat{e}(S_t, U_0) \cdot \prod_{i=2}^t \hat{e}(Q_{i-1}, U_i)^{-1} &= \hat{e}\left(\sum_{i=1}^t [s_{i-1}]P_i, [t]P_0\right) \cdot \prod_{i=2}^t \hat{e}([s_{i-1}]P_0, [t]P_i)^{-1} \\
 &= \hat{e}\left(\sum_{i=1}^t [s_{i-1}]P_i, [t]P_0\right) \cdot \prod_{i=2}^t \hat{e}(-[s_{i-1}]P_i, [t]P_0) \\
 &= \hat{e}\left(\sum_{i=1}^t [s_{i-1}]P_i, [t]P_0\right) \cdot \hat{e}\left(-\sum_{i=2}^t [s_{i-1}]P_i, [t]P_0\right) \\
 &= \hat{e}([s_0]P_1, [t]P_0) \\
 &= \hat{e}(P_1, [s_0]P_0)^t \\
 &= \hat{e}(P_1, Q_0)^t.
 \end{aligned}$$

A few comments on this scheme are in order. Firstly, note that encryption only requires one pairing computation, and this needs only to be computed once to enable communication with any entity registered in the hierarchy. On the other hand, t pairing computations are required for every decryption. It would be interesting to find hierarchical schemes with an alternative balance between the costs of encryption and decryption. Secondly, notice how the length of ciphertexts grows with t – this seems inescapable in a hierarchical system. Thirdly, note that the scheme has a strong in-built escrow, in that any ancestor of an entity can decrypt ciphertexts intended for that entity: an ancestor at level j can use the equation

$$M' = V \oplus H_2 \left(\hat{e}(S_j, U_0) \cdot \prod_{i=2}^j \hat{e}(Q_{i-1}, U_i)^{-1} \right)$$

to decrypt a message encrypted for a child at level t .

X.5.2. Extensions of the Basic Scheme. In [147], Gentry and Silverberg also showed how to use the techniques of Fujisaki–Okamoto [128] to produce a strengthened encryption scheme which is secure against chosen-ciphertext attackers in the random oracle model, provided that the BDH problem is hard. The security model adopted in [147] is sufficiently strong to capture collusions of entities attempting to compromise the private keys of their ancestors. This is because it allows the adversary to extract the private keys of entities at any level in the hierarchy and to adaptively select the identity on which it wishes to be challenged.

Naor’s idea for turning an IBE scheme into a signature scheme was exploited in [147] to produce a hierarchical identity-based signature (HIBS) scheme. The security of this scheme depends on the hardness of the CDH problem in G_1 . Gentry and Silverberg also considered how the NIKDS of Sakai et al. can be used to reduce the amount of computation needed for

encryption between two parties who are “near” to one another in the hierarchy. The resulting scheme also enjoys shorter ciphertexts. A number of other variants on this theme are also explored in [147].

X.5.3. Related Topics. Canetti, Halevi and Katz [55] built upon the work of [147] to produce the first non-trivial forward-secure public-key encryption (FS-PKE) scheme. In an FS-PKE scheme, a user has a fixed public key but a private key which evolves over time; such a scheme should then have the property that a compromise of the user’s private key at time t does not affect the security of messages encrypted during earlier time periods (though clearly no security can be guaranteed after time t).

The scheme in [55] makes use of a basic primitive called a binary tree encryption (BTE) scheme. A BTE scheme consists of a single “master” public key, a binary tree of private keys together with encryption and decryption algorithms and a routine which computes the private keys of the children of a node from the private key at that node. The encryption algorithm takes as input the public key and the label of a node. A selective-node chosen-ciphertext attack (SN-CCA) against a BTE scheme goes roughly as follows. The adversary selects a target node to attack in the challenge phase in advance. The adversary is then given the private keys for a certain set of nodes. This set consists of all the children of the target together with all the siblings of the target’s ancestors. This is the maximal set of private keys which the adversary can be given without enabling the trivial computation of the private key of the target node. The adversary’s job is then to distinguish ciphertexts encrypted under the public key and target node, given access to a decryption oracle.

Canetti, Halevi and Katz show how a BTE scheme secure against SN-CCA attacks can be constructed from a simplification of the HIBE scheme of [147]. They then show how any SN-CCA secure BTE scheme can be used in a simple construction to obtain an encryption scheme that is forward-secure in a natural adaptation of the standard IND-CCA2 model for public-key encryption. The trick is to traverse the tree of the BTE in a pre-order traversal, with the key at the t th node in the traversal determining how the private key in the forward-secure scheme is updated at time t . The security definition for a BTE scheme quickly converts into the desired forward security. Combining their constructions, the authors of [55] obtain an efficient, forward-secure encryption scheme whose security rests of the hardness of the BDH problem in the random oracle model.

A BTE scheme secure in the SN-CCA sense, but without requiring random oracles, is also constructed in [55]. The construction uses $O(\ell)$ -wise independent hash functions, and the security of the resulting BTE scheme depends on the hardness of the DBDH problem rather than the BDH problem. However the construction gives a completely impractical scheme because of its reliance on non-interactive zero-knowledge proofs. As an interesting aside, Canetti,

Halevi and Katz go on to show how an HIBE scheme can be constructed from a BTE scheme, though with a weaker security model than is considered in [147]. A corollary of this result is the construction of an IBE scheme (and an HIBE scheme) that is secure in the standard model (i.e., without the use of random oracles) assuming the hardness of the DBDH problem, though only for an adversary who specifies in advance which identity he will attack. Again the scheme will be impractical if it is to be secure against chosen-ciphertext attacks.

One issue that the proofs of security in [55] have in common with those of [39, 147] (and indeed many papers in the area) is that the security reductions are not particularly tight. For example, a factor of $1/N$ is introduced in [55, Proof of Theorem 4], where N is the number of time periods supported by the FS-PKE scheme. It seems to be a challenging problem to produce results tightly relating the security of the schemes to the hardness of some underlying computational problems.

Canetti, Halevi and Katz [56] have shown a surprising connection between IBE and chosen-ciphertext security for (normal) public-key encryption. They give a construction for an IND-CCA2 secure scheme of the latter type from a weakly-secure IBE scheme and a strongly unforgeable one-time signature scheme. Here, the IBE scheme need only be secure against chosen-plaintext attacks by selective-ID adversaries, that is, adversaries who specify in advance which identity they will attack in the challenge phase. The twist needed to make the construction work is to interpret the public key of the signature scheme as an identity in the IBE scheme, for which the decrypting party holds the master secret. Since a weakly-secure IBE scheme can be constructed in the standard model, the results of [56] yield a new IND-CCA2 secure public-key encryption scheme whose security does not rely on the random oracle assumption.

Boneh and Boyen [36] provided new and efficient constructions for an HIBE scheme and an IBE scheme using pairings. Both schemes are secure in the standard model, against selective-ID, chosen plaintext attackers. The HIBE scheme is secure given that the DBDH problem is hard. It can be converted into a selective-ID, chosen-ciphertext secure HIBE scheme using the method of [56]; the resulting scheme is efficient. The security of the new IBE scheme in [36] depends on the hardness of a new problem, the decision bilinear Diffie–Hellman Inversion problem (DBDHI problem), which is related to a decisional version of the k -weak CDH problem of [250]. This scheme is also closely related to the signature scheme of [35]. Unfortunately, no efficient conversion to a chosen-ciphertext secure scheme is currently known. However, by combining this scheme with ideas in [56] and the signature scheme of [35], one obtains a reasonably efficient public-key encryption scheme that is IND-CCA2 secure in the standard model.

Forward secure encryption is perhaps the most basic form of what might be called “key updating cryptography.” Here the general approach is to have an evolving private key which may or may not be updated with the help of a second entity called a base or helper. Several other papers use pairings to address problems in this area. Of particular note is the work of Bellare and Palacio in [23] and of Dodis et al. in [107]. In the former paper, the authors construct a strongly key-insulated encryption scheme from the IBE scheme of Boneh and Franklin. Such a scheme allows a user to cooperate with a helper to refresh his private key; the scheme remains secure even if the user’s private key is corrupted in up to some threshold number of time periods, and even if the helper is compromised (so long as the user’s key then is not). Bellare and Palacio also provide an equivalence result in [23, Theorem 4.1], relating the existence of a secure IBE scheme to that of a secure strongly key-insulated encryption scheme. Dodis et al. [107] work with an even stronger security model, in which the base can also be frequently corrupted, and construct an intrusion-resilient public-key encryption scheme from the forward-secure scheme of [55].

Yum and Lee [352] have explored similar concepts in the context of signatures, using the IBS scheme of [62] to obtain efficient key updating signature schemes.

X.6. More Key Agreement Protocols

Alongside encryption and signatures, key agreement is one of the fundamental cryptographic primitives. As we have already seen in Section X.2, pairings were used early on to construct key agreement schemes and protocols. In this section we examine how this area has developed since the foundational work of [284, 183].

X.6.1. Two-Party Key Agreement Protocols. The NIKDS of Sakai et al. [284] allows two parties to non-interactively agree the identity-based key $K_{AB} = \hat{e}(Q_A, Q_B)^s$ after they have registered with the same TA and obtained their respective private keys $S_A = [s]Q_A$, $S_B = [s]Q_B$. However, the key K_{AB} is a static one, while many applications require a fresh key for each communications session.

Smart [312] was the first author to consider how pairings could be used to develop identity-based, authenticated key agreement protocols. His protocol uses the same keying infrastructure as the IBE scheme of Boneh and Franklin. In particular, system parameters $\langle G_1, G_3, \hat{e}, P, Q_0 = [s]P, H_1 \rangle$ are pre-established and entities A , B possess private keys $S_A = [s]Q_A$, $S_B = [s]Q_B$. Here, $Q_A = H_1(\text{ID}_A)$, where ID_A is the identity string of A . Q_B is defined similarly. In Smart’s protocol, A and B exchange ephemeral values $T_A = [a]P$ and $T_B = [b]P$, where a , b are selected at random from $\mathbb{Z}_p^*$. Notice that these are identical to the messages exchanged in a straightforward

Diffie–Hellman protocol for the group G_1 . Entity A then computes:

$$K_A = \hat{e}([a]Q_B, Q_0) \cdot \hat{e}(S_A, T_B)$$

while entity B computes:

$$K_B = \hat{e}([b]Q_A, Q_0) \cdot \hat{e}(S_B, T_A).$$

It is an easy exercise to show that

$$K_A = K_B = \hat{e}([a]Q_B + [b]Q_A, [s]P)$$

so that this common value can be used as the basis of a shared session key. The bandwidth requirements of the protocol are moderate, being one element of G_1 per participant. A version of the basic protocol offering key confirmation is also considered in [312]: this service ensures that each entity gets a guarantee that the other entity actually has calculated the shared key. While no attacks have been found on this protocol to date, no formal security analysis has been given either.

Smart’s protocol requires two pairing computations per participant. An alternative protocol was given by Chen and Kudla in [71]. In their protocol, A and B exchange ephemeral values $W_A = [a]Q_A$ and $W_B = [b]Q_B$ and compute the keys

$$K_A = \hat{e}(S_A, W_B + [a]Q_B), \quad K_B = \hat{e}(W_A + [b]Q_A, S_B).$$

Now $K_A = K_B = \hat{e}(Q_A, Q_B)^{s(a+b)}$ can be computed using just one pairing operation. A useful security model that is applicable for this type of protocol is the extension of the Bellare–Rogaway model [25] to the public key setting that was developed by Blake-Wilson et al. in [28, 29]. It is proved in [70] that the above protocol is a secure authenticated key agreement in this model, provided the BDH problem is hard. The original proof of this result published in [71] is flawed, and a strong restriction on adversarial behaviour is needed to provide the corrected version in [70]. Chen and Kudla also consider modifications of their protocol which provide forward secrecy, anti-escrow features and support for multiple TAs.

Other authors have also tried to adapt Smart’s protocol. Shim’s attempt [302] was shown to be vulnerable to a man-in-the-middle attack in [322]. Yi’s protocol [351] halves the bandwidth required by Smart’s protocol using a form of point compression.

An alternative approach to identity-based key agreement was taken by Boyd et al. in [47]. In this work the non-interactively agreed key $K_{AB} = \hat{e}(Q_A, Q_B)^s$ of Sakai et al. is used as the key to a MAC algorithm to provide authentication of the messages in a Diffie–Hellman key exchange. The resulting protocol is provably secure in the model developed in [22, 57] and has the interesting privacy feature of providing deniable authentication: since either party could have computed all the messages in a protocol run, both parties can also deny having taken part in the protocol. The authors of [47] also considered the use of identity-based encryption as a session key transport

mechanism. Related uses of the key $\hat{e}(Q_A, Q_B)^s$ in “secret handshake” key agreement protocols were also explored in [13], where the integration of these protocols into the SSL/TLS protocol suite was also studied.

X.6.2. Multi-Party Key Agreement Protocols. In this section we discuss how Joux’s protocol [183] has inspired new protocols for multi-party key agreement.

Recall that in Joux’s protocol, the key agreed between three parties is equal to $\hat{e}(P, P)^{abc}$ when the ephemeral broadcast values are $[a]P$, $[b]P$ and $[c]P$. We have noted in Section X.2.2 that this protocol is vulnerable to man-in-the middle attacks because it is not authenticated. An obvious way to enhance the security of the protocol is to add signatures to the ephemeral values. A number of efficient, signature-free approaches to securing Joux’s protocol were described in [8]. It was also shown in [8], perhaps surprisingly, that an authenticated version of Joux’s protocol has no benefit over a simple extension of the Diffie–Hellman protocol when three-party, authenticated protocols with confirmation are considered in a non-broadcast environment: any secure protocol will require at least six messages in this context. Galbraith *et al.* [135] have studied the bit security of the BDH problem; their results can be applied to Protocols of [8] and [312] to show that it is secure to use a finite-field trace operation to derive a session key from the raw key material exchanged in these protocols.

Shim’s attacks [300] on the protocols of [8] show that adding authentication to three-party protocols is a delicate business. Zhang and Liu [356] developed identity-based, authenticated versions of Joux’s protocol.¹⁵ Nalla and Reddy [259] also put forward identity-based, three-party key agreement protocol, but these were all broken in [74, 299]. Meanwhile, Shim’s proposal for a three-party protocol [301] was broken in [322].¹⁶

Protocols for more than three parties, using Joux’s protocol and its derivatives as a building block, have been considered by several authors [281, 14]. Lack of space prevents their detailed consideration here. For attacks on some other schemes which attempted to mimic the Burmester-Desmedt protocol of [53], see [353].

X.7. Applications and Infrastructures

It should be apparent that one of the major uses of pairings has been in developing identity-based cryptographic primitives. So far, we have said little about what identity-based public-key cryptography (ID-PKC) has to

¹⁵Note that there is no real benefit in deriving eight different keys from a single key exchange by algebraic manipulations as in [356]: a simple key derivation function based on hashing suffices.

¹⁶Even though the protocol defined in [301] does not actually make mathematical sense! For it involves an exponentiation of an element $\hat{e}(P, P)$ in G_3 to a power that is a product of an element in $\mathbb{Z}_r^*$ and an element in G_3 .

offer in comparison to more traditional forms of public-key cryptography. We rectify this in the first part of this section. We go on to study how pairings have been used to develop new architectures supporting the deployment of public-key cryptography. Then in the third part we outline a variety of recent work in which pairings have been put into practice, either in trials of identity-based technology or in on-paper proposals outside the immediate confines of cryptography.

X.7.1. Further Development of Identity-based Systems. We introduced the concepts of identity-based encryption (IBE) and, more generally, ID-PKC in Sections X.2.1 and X.3, portraying them as being useful alternatives to traditional PKIs. Here we explore in a little more detail why this is the case and critically examine some of the problems inherent in identity-based approaches.

X.7.1.1. Identity-Based Systems Versus Traditional PKIs. Recall that in an identity-based system, a TA is responsible for issuing private keys to the correct users. This TA in effect replaces the CA in a traditional PKI, but the roles of TA and CA are somewhat different. The CA in a traditional PKI does not usually know users' private keys, but rather issues certificates which assert a binding between identities and public keys. The TA in an identity-based system is responsible for checking that applicants do have the claimed identity and then issuing the corresponding private key. Thus identity-based systems automatically have a key escrow facility. Whether this is a good thing or not will depend on the particular application at hand. It will certainly be a useful feature in many "corporate" deployment scenarios, where the recovery of encrypted files and e-mail may well be important should an employee leave the organization, say. However, escrow can complicate the issue of non-repudiation of signatures. For example, an important piece of EU legislation [EU 1999] requires that the signing key be under the sole control of the signing party in order that a signature be recognized as an "advanced electronic signature." Thus traditional signatures supported by a PKI are likely to be more useful than identity-based signatures in practice.

Note that, in both ID-PKC and traditional PKI, it is important to authenticate applicants before issuing valuable data (private keys in the former, certificates in the latter). So some additional authentication mechanism is needed at the time of registration/key issuance. Both systems also require that any system parameters (e.g., a root certificate or a TA's public parameters) are authentically available to users. However, with ID-PKC, there is an additional requirement: the private keys must be delivered over confidential and authentic channels to the intended recipients. Again this seems to point towards the enterprise as being a fruitful deployment area for ID-PKC

– for example, one could use a company’s internal mail system and personnel database to distribute keys and control registration for low-to-medium security applications.

The particular IBE scheme of Boneh and Franklin [39] supports multiple TAs and split private keys in a very natural way. This goes some way to addressing escrow concerns. For example, suppose two TAs share parameters $\langle G_1, G_3, \hat{e}, P \rangle$ but have master secrets $s_1, s_2 \in \mathbb{Z}_r^*$ and public values $Q_1 = [s_1]P$, $Q_2 = [s_2]P$. Then a user A with identity string ID_A can form his private key as the sum $[s_1]Q_A + [s_2]Q_A = [s_1 + s_2]Q_A$ of the private keys obtained from each TA. To encrypt to A , ciphertexts of the form

$$\langle [t]P, M \oplus H_2(\hat{e}(Q_A, Q_1 + Q_2)^t) \rangle$$

can be used. More generally, a k -out-of- n escrow capability can be established – see [39] for details. Such a facility is also supported by many other ID-based schemes developed post-Boneh–Franklin.

The ability to make use of multiple TAs was exploited in [69] to create cryptographic communities of interest. Here, each TA represents a particular group (e.g., the group of all people having the same citizenship, profession or name); a sum of keys from different groups creates intersections of groups all of whose members can calculate the same private key.

Another point of comparison for traditional public key and ID-PKC systems is the issue of revocation. Whenever a certificate in a traditional system expires (perhaps because the end of its validity period is reached or because of a private key compromise), this fact must be communicated to the parties relying on the certificates. There is the same requirement for timely transmission of revocation information in an ID-PKC system too. It has been suggested by many authors that in ID-PKC, one can simply attach a validity period to identities, for example “john.smith || 2004”, so that public keys automatically expire. However, such a system is no longer purely identity-based, and one must still find a way to deal with keys that become compromised before the end of their expiry period.

A deeper comparison of revocation and many other issues for ID-PKC and traditional PKIs is made in [274]. Whether ID-PKC really has something to offer over traditional PKIs and even symmetric systems very much depends on the application context, on what is to be secured and on what constraints there are on the solutions that can be adopted. It is certainly not the case that an identity-based approach will be the correct one in every circumstance.

X.7.1.2. Cryptographic Workflows. An apparently innocuous feature of IBE is that when encrypting a message for entity A , the sender can choose the identity string ID_A used in the encryption process. Only if A has the matching private key $[s]Q_A = [s]H_1(\text{ID}_A)$ will he be able to decrypt the message. Naturally, in many situations, it is most convenient if the sender chooses a string ID_A for which this is the case. However, it is possible that

A 's identity ID_A and public key Q_A are actually determined *before* the private key $[s]Q_A$. This can have interesting consequences. For example, the sender can encode in A 's identity string a set of conditions (or a policy) that should be met before the TA, acting as a policy monitor, should issue the private key.

The idea of encoding conditions in identity strings can be combined with the use of multiple TAs to create a *cryptographic workflow*, that is, a sequence of private key issuances that must be successfully carried out before an entity can decrypt a ciphertext. In this context, the “I” in ID-PKC is better interpreted as “identifier”, since rarely will identities be used alone.

As an example of this concept in action, consider the scenario where a customer wants his bank manager to have access to a particular instruction, but only after a certain time. Suppose the bank acts as a TA for its employees in a Boneh–Franklin IBE scheme with the usual parameters $\langle G_1, G_3, \hat{e}, P \rangle$, master secret s_{bank} and public parameter $Q_{bank} = [s_{bank}]P$. Suppose that the bank manager has received his private key $[s_{bank}]H_1(ID_{bm})$. Suppose also that a third party operates an encrypted time service as follows. The third party, using the same basic public parameters as the bank, acts as a TA with master secret s_{time} and public parameter $Q_{time} = [s_{time}]P$. At time T , the third party broadcasts to all subscribers the private key $[s_{time}]H_1(T)$. Now, to encrypt an instruction M for the bank manager to be read only after time T_0 , the customer creates the ciphertext:

$$C = ([t]P, M \oplus H_2(\hat{e}(Q_{bank}, H_1(ID_{bm}))^t \cdot \hat{e}(Q_{time}, H_1(T_0))^t)).$$

Here, the customer has encrypted M using both the identity of the bank manager and the time T_0 after which the message is to become decryptable. Only after time T_0 can the bank manager access the value $[s_{time}]H_1(T_0)$ and combine this with his private key $[s_{bank}]H_1(ID_{bm})$ in the bank's scheme to compute the value:

$$H_2(\hat{e}([t]P, [s_{bank}]H_1(ID_{bm})) \cdot \hat{e}([t]P, [s_{time}]H_1(T_0))),$$

allowing the decryption of the ciphertext C .

In this example, the customer created a special public key for encryption out of two identifiers, the bank manager's identity and the time identifier. These identifiers come from two different schemes with two different TAs, but ones who share some parameters – perhaps they are using standardized groups and pairings.¹⁷ The customer has used multiple TAs to create a workflow that the bank manager must follow in order to access the desired information: first the bank manager must obtain his private key in the bank's scheme; then he must wait for the time service to reveal the private key at time T_0 .

It is easy to imagine other scenarios where the dynamic creation of workflows in this way could be very useful. There is no theoretical limit on the

¹⁷In fact, the reliance on shared parameters can be almost completely eliminated by slightly modifying the encryption algorithm.

number of private keys that the recipient must fetch or the types of roles or identifiers that can be used. The recipient may be required to perform some kind of authentication (based on identity, address, role in an organization, etc.) at each stage. Further research along these lines, allowing the expression of more complex conditions in identifiers, can be found in [69, 314].

X.7.2. New Infrastructures. Some form of hierarchy seems necessary in order to address the scalability and availability issues inherent in any system with a single point of distribution for keying material. We have seen how the work of Gentry and Silverberg [147] allows a hierarchy of TAs in ID-based systems. Chen et al. [68] have studied the benefits of developing a mixed architecture, with identity-based TAs administering users at the lowest levels of the hierarchy being supported by a traditional PKI hierarchy above.

In [146], Gentry introduced the concept of Certificate-Based Encryption (CBE), with a view to simplifying revocation in traditional PKIs, and used pairings to construct a concrete CBE scheme. We give a brief review of Gentry's scheme using notation as previously established: P generates G_1 of prime order r , $\hat{e} : G_1 \times G_1 \rightarrow G_3$ is a bilinear map and $H_2 : G_3 \rightarrow \{0, 1\}^n$ is a hash function.

In Gentry's CBE scheme, an entity A 's private key consists of two components. The first component $[s_C]P_A(i)$ is time-dependent and is issued as a certificate to A on a regular basis by a CA. Here s_C is the CA's private key and $P_A(i) \in G_1$ is derived from hashing certain parameters, including A 's public key $[s_A]P$ and the current time interval i . The second component $[s_A]P'_A$ is chosen by A and kept private. Here, $P'_A \in G_1$ is derived from A 's identifying data. So A 's private key is the sum $[s_C]P_A(i) + [s_A]P'_A$, a time-dependent value that is only available to A if A is certified in the current time interval. Now, to encrypt a message M for A , an entity selects t at random from $\mathbb{Z}_r^*$ and sets:

$$C = \langle [t]P, M \oplus H_2(\hat{e}([s_C]P, P_A(i))^t \cdot \hat{e}([s_A]P, P'_A)^t) \rangle.$$

Notice that $[s_C]P$ is available to encrypting parties as a public parameter of the CA, while $P_A(i)$, P'_A can be computed from A 's public information and $[s_A]P$ is A 's public key. Decryption by A is straightforward if A has $[s_C]P_A(i)$. For if $C = \langle U, V \rangle$, then A can compute:

$$\begin{aligned} \hat{e}(U, [s_C]P_A(i) + [s_A]P'_A) &= \hat{e}([t]P, [s_C]P_A(i)) \cdot \hat{e}([t]P, [s_A]P'_A) \\ &= \hat{e}([s_C]P, P_A(i))^t \cdot \hat{e}([s_A]P, P'_A)^t. \end{aligned}$$

Notice that the private key $[s_C]P_A(i) + [s_A]P'_A$ used here can be regarded as a two-party aggregate signature in the scheme of [40]. The second private component $[s_C]P_A(i)$ acts as an *implicit certificate* for relying parties, one that a relying party can be assured is only available to A provided that A 's certificate has been issued for the current time period by the CA. The security of CBE depends critically on the CA binding the correct public key into A 's

implicit certificate in each time period. Thus (quite naturally), the initial registration of users and their public keys must take place over an authentic channel and be bootstrapped from some other basis for trust between A and the CA.

This approach can significantly simplify revocation in PKIs. For notice that there is no need to make any status checks on A 's public key before encrypting a message for A . So there is no requirement for either Certificate Revocation Lists or an on-line certificate status checking protocol. However, the basic CBE approach of [146] does have a major drawback: the CA needs to issue new values $[s_C]P_A(i)$ to every user in the scheme in every time period. A granularity of one hour per time period is suggested in [146]; this substantially adds to the computation and communication that take place at the CA for a PKI with even a small user base. The basic CBE approach can be regarded as effectively trading simplified revocation for an increased workload at the CA. A number of enhancements to the basic CBE approach are also presented in [146]. These reduce the work that must be carried out by the CA.

A security model for CBE is also developed in [146], and Gentry goes on to show that the CBE scheme described above, but modified using the Fujisaki-Okamoto technique [128], meets the definition of security for the scheme, provided that the BDH problem is hard. It is clear that similar ideas to Gentry's can be applied to produce certificate-based signature schemes. A scheme of this type was developed in [192].

Al-Riyami and Paterson [9] proposed another new model for supporting the use of public-key cryptography which they named certificateless public-key cryptography (CL-PKC). Independently, Chen et al. [73] proposed similar ideas in the context of signatures and group signatures. The key feature of the model of [9] is that it eliminates the need for certificates, hence the (somewhat clumsy) adjective "certificateless."

Pairings are used to construct concrete CL-PKC schemes in [9]. As in [146], an entity A 's private key is composed in two stages. Firstly, an identity-dependent *partial private key* $[s]Q_A = [s]H_1(\text{ID}_A)$ is received over a confidential and authentic channel from a trusted authority (called a key generation centre, KGC).¹⁸ Secondly, A combines the partial private key $[s]Q_A$ with a secret x_A to produce his private key $S_A = [x_A s]Q_A$. The corresponding public key is the pair $\langle X_A, Y_A \rangle = \langle [x_A]P, [x_A]Q_0 \rangle$, where $Q_0 = [s]P$ is a public parameter of the system. The certificateless encryption (CL-PKE) scheme of [9] is obtained by adapting the IBE scheme of Boneh and Franklin [39] and operates as follows in its basic form. To encrypt a message for A , an entity

¹⁸This partial private key $[s]H_1(\text{ID}_A)$ is identical to the private key in the IBE scheme of Boneh and Franklin. It can also be regarded as a BLS signature by the TA on A 's identity, and hence as a form of certification, though one that does not involve A 's public key.

first checks that the equality

$$\hat{e}(X_A, Q_0) = \hat{e}(Y_A, P)$$

holds, then selects t at random from $\mathbb{Z}_r^*$ and sets:

$$C = \langle [t]P, M \oplus H_2(\hat{e}(Q_A, Y_A)^t) \rangle.$$

It is easy to see that, to decrypt $C = \langle U, V \rangle$, A can use his private key $S_A = [x_A s]Q_A$ and compute $M = V \oplus H_2(\hat{e}(S_A, U))$.

Notice that, in this encryption scheme, A 's public key need not be supported by a certificate. Instead, an entity A who wishes to rely on A 's public key is assured that, if the KGC has done its job properly, only A who is in possession of the correct partial private key and user-generated secret could perform the decryption. Because there are no certificates, Al-Riyami and Paterson [9] were forced to consider a security model in which the adversary is allowed to replace the public keys of entities at will. The security of the scheme then rests on the attacker not knowing the partial private keys. Security against the KGC is also modelled in [9], by considering an adversary who knows the master secret s for the scheme, but who is trusted not to replace the public keys of entities. The security of the encryption scheme in [9] rests on the hardness of a new problem generalising the BDH problem:

Generalized Bilinear Diffie–Hellman Problem (GBDH Problem):

Given P , $P_1 = [a]P$, $P_2 = [b]P$ and $P_3 = [c]P$ in G_1 with a , b and c selected uniformly at random from $\mathbb{Z}_r^*$, output a pair

$$Q, \quad \hat{e}(P, Q)^{abc}$$

where $Q \in G_1$.

Al-Riyami and Paterson [9] also present certificateless signature, key exchange and hierarchical schemes. These are obtained by adapting schemes of [164, 312, 147]. CL-PKC supports the temporal re-ordering of public and private key generation in the same way that ID-PKC does, thus it can be used to support workflows of the type discussed in Section X.7.1.2.

CL-PKC combines elements from ID-PKC and traditional PKI. On the one hand, the schemes are no longer identity-based: they involve the use of A 's public key, which is no longer simply derived from A 's identity. On the other hand, CL-PKC avoids the key escrow inherent in ID-PKC by having user-specific private information involved in the key generation process. CL-PKC does not need certificates to generate trust in public keys; instead, this trust is produced in an implicit way. This would appear to make CL-PKC ideal for systems where escrow is unacceptable but where the full weight of PKI is untenable.

There is a close relationship between the ideas in [146] and [9]. It is possible to convert CL-PKE scheme into a CBE scheme: if A 's identity in the CL-PKE scheme is extended to include a time period along with the public key, then the CL-PKE scheme effectively becomes a CBE scheme. On

the other hand, if one omits certain fields from the certificates in a CBE scheme, one obtains an encryption scheme that is functionally similar to a CL-PKE scheme. Differences do remain: in the strength and scope of the two security models developed in [146] and [9], as well as in the technical details of the schemes' realizations.

X.7.3. Applications and Implementations. In this section we provide brief notes on recent work putting pairings into practice or using pairings in the broader context of information security.

A number of authors have examined how pairings can be put to use to enhance network security. Kempf et al. [197] described a lightweight protocol for securing certain aspects of IPv6. The protocol adds identity-based signatures to router and neighbour advertisements, with identities being based on IP addresses. Khalili et al. [198] combined identity-based techniques with threshold cryptography to build a key distribution mechanism suitable for use in ad hoc networks.

Appenzeller and Lynn [10] proposed using the NIKDS of Sakai et al. [284] to produce identity-based keys for securing IP packets between hosts. Their approach adds security while avoiding the introduction of state at the network layer, and so provides an attractive alternative to IPSec. However, it can only be used by pairs of entities who share a common TA. On the other hand, Smetters and Durfee [315] proposed a system in which each DNS domain runs its own IBE scheme and is responsible for distributing private keys to each of its hosts (or e-mail users). Inter-domain IPSec key exchanges and e-mail security are enabled by extending DNS to give a mechanism for distributing IBE scheme parameters. In [315], a protocol of [70] is used to provide an alternative to IKE (IPSec Key Exchange) for inter-domain exchanges while the NIKDS of Sakai et al. [284] can be used to set up IKE in pre-shared key mode for intra-domain communications. The protocol resulting in the latter case in [315] is similar to a protocol proven secure in [47].

Dalton [94] described the particular computing and trust challenges faced in the UK's National Health Service and studied the applicability of identity-based techniques in that environment.

Waters et al. [345] modified the IBE scheme of Boneh and Franklin [39] to provide a solution to the problem of searching through an encrypted, sensitive audit log. In the scheme of [345], a machine attaches a set of IBE-encrypted tags to each entry in its log, each tag corresponding to a single keyword W . The "identity" used in the encryption to produce a tag is the string W , while the plaintext encrypted is the symmetric key that was used to encrypt the entry in the log (plus some redundancy allowing the plaintext to be recognized). The TA for the IBE system acts as an audit escrow agent: when an entity requests the capability to obtain log entries containing a particular keyword, the TA may provide the private key $[s]H_1(W)$ matching that keyword. Now the testing entity can simply try to

decrypt each tag for the log entry. When the correct tag is decrypted, a key allowing the entry to be decrypted results. A more theoretical and formal approach to the related problem of searchable public-key encryption (SPKE) can be found in [37]. One of the three constructions for an SPKE scheme in [37] is based on pairings, specifically, it is again an adaptation of the IBE scheme of Boneh and Franklin.

Currently, we know of at least one company, Voltage Security, that is actively developing and marketing identity-based security systems. Their products include secure e-mail and file encryption applications. An early identity-based secure e-mail demonstrator, implementing Boneh and Franklin's IBE scheme, is still available from

<http://crypto.stanford.edu/ibe/download.html>

at the time of writing. Routines for Weil and Tate pairing computations are built into a number of software libraries, including Magma.

X.8. Concluding Remarks

We have seen in this chapter how pairings have been used to build some entirely new cryptographic schemes and to find more efficient instantiations of existing primitives. Although we have not been exhaustive in our coverage, we trust that the breathless pace of research in the area is apparent. What might the future hold for this subject, and what are the most important questions yet to be tackled?

The techniques and ideas used in pairing-based cryptography are very new, so it is hard to envisage where they will be taken next. The applications in topics like intrusion-resilient encryption and cryptographic workflows are so surprising (at least to the author) that accurately predicting an answer to the first question seems fraught. One might expect the rate of publication of new pairing-based schemes to slow a little and a period of consolidation to occur. On a more theoretical note, the subject is rife with random oracles and inefficient reductions. Removing these whilst keeping the full strength of the security models and obtaining practical schemes should keep cryptographers busy.

We suggest that much more work above and below the purely cryptographic level is needed.

As Section X.7.3 illustrates, techniques from pairing-based cryptography are beginning to have an effect on other domains of information security. Attempts at commercialization will provide a true test of the applicability of what on paper seem like very neat ideas. Identity-based cryptography is certainly interesting, but it still has much to prove when measured against traditional PKIs. One topic we have not addressed here is that of intellectual property and patents. This may become a major factor in the take-up of the technology, in the same way that it was for elliptic curve cryptography in the last decade and public-key cryptography before that.

Below the cryptographic level, more work on the fundamental question of understanding the hardness of the BDH problem (and the associated decisional problem) seems essential. While the relationships to the CDH problem and other problems in related groups are well understood, this is of course not the whole story. Pairings also give new relevance to “old” problems, for example, evaluating the performance of discrete logarithm algorithms in fields of small characteristic for concrete parameters. One might also worry about relying too much on the extremely narrow class of supersingular curves for constructing pairings. This is akin to the days before point counting for curves of cryptographic sizes became routine, when CM curves were suggested as a way of proceeding. It is interesting to note that recent constructions for curves with prescribed embedding degrees (as described in Chapter IX) also rely on CM methods, while it is known that the embedding degree of a random curve of a particular size will be very high. The challenge to computational number theorists is evident.

Bibliography

- [ECC] I.F. Blake, G. Seroussi and N.P. Smart. *Elliptic Curves in Cryptography*. Cambridge University Press, 1999.
- [EP] IACR e-print archive. Available from <http://eprint.iacr.org/>.
- [A-1] L. Adleman and M.-D. Huang, editors. *ANTS-1: Algorithmic Number Theory*. Springer-Verlag, LNCS 877, 1994.
- [A-2] H. Cohen, editor. *ANTS-2: Algorithmic Number Theory*. Springer-Verlag, LNCS 1122, 1996.
- [A-3] J.P. Buhler, editor. *ANTS-3: Algorithmic Number Theory*. Springer-Verlag, LNCS 1423, 1998.
- [A-4] W. Bosma, editor. *ANTS-4: Algorithmic Number Theory*. Springer-Verlag, LNCS 1838, 2000.
- [A-5] C. Fieker and D.R. Kohel, editors. *ANTS-5: Algorithmic Number Theory*. Springer-Verlag, LNCS 2369, 2002.
- [A-6] D. Buell, editor. *ANTS-6: Algorithmic Number Theory*. Springer-Verlag, LNCS 3076, 2004.
- [A98] K. Ohta and D. Pei, editors. *Advances in Cryptology – ASIACRYPT '98*. Springer-Verlag, LNCS 1514, 1998.
- [A99] K.Y. Lam, E. Okamoto and C. Xing, editors. *Advances in Cryptology – ASIACRYPT '99*. Springer-Verlag, LNCS 1716, 1999.
- [A00] T. Okamoto, editor. *Advances in Cryptology – ASIACRYPT 2000*. Springer-Verlag, LNCS 1976, 2000.
- [A01] C. Boyd, editor. *Advances in Cryptology – ASIACRYPT 2001*. Springer-Verlag, LNCS 2248, 2001.
- [A02] Y. Zheng, editor. *Advances in Cryptology – ASIACRYPT 2002*. Springer-Verlag, LNCS 2501, 2002.
- [A03] C.S. Lai, editor. *Advances in Cryptology – ASIACRYPT 2003*. Springer-Verlag, LNCS 2894, 2003.
- [C84] G.R. Blakley and D. Chaum, editors. *Advances in Cryptology – CRYPTO '84*. Springer-Verlag, LNCS 196, 1985.
- [C89] G. Brassard, editor. *Advances in Cryptology – CRYPTO '89*. Springer-Verlag, LNCS 435, 1990.
- [C91] J. Feigenbaum, editor. *Advances in Cryptology – CRYPTO '91*. Springer-Verlag, LNCS 576, 1992.
- [C92] E.F. Brickell, editor. *Advances in Cryptology – CRYPTO '92*. Springer-Verlag, LNCS 740, 1993.
- [C93] D. Stinson, editor. *Advances in Cryptology – CRYPTO '93*. Springer-Verlag, LNCS 773, 1993.
- [C96] N. Koblitz, editor. *Advances in Cryptology – CRYPTO '96*. Springer-Verlag, LNCS 1109, 1996.
- [C97] B.S. Kaliski Jr., editor. *Advances in Cryptology – CRYPTO '97*. Springer-Verlag, LNCS 1294, 1997.

- [C98] H. Krawczyk, editor. *Advances in Cryptology – CRYPTO '98*. Springer-Verlag, LNCS 1462, 1998.
- [C99] M. Wiener, editor. *Advances in Cryptology – CRYPTO '99*. Springer-Verlag, LNCS 1666, 1999.
- [C00] M. Bellare, editor. *Advances in Cryptology – CRYPTO 2000*. Springer-Verlag, LNCS 1880, 2000.
- [C01] J. Kilian, editor. *Advances in Cryptology – CRYPTO 2001*. Springer-Verlag, LNCS 2139, 2001.
- [C02] M. Yung, editor. *Advances in Cryptology – CRYPTO 2002*. Springer-Verlag, LNCS 2442, 2002.
- [C03] D. Boneh, editor. *Advances in Cryptology – CRYPTO 2003*. Springer-Verlag, LNCS 2729, 2003.
- [CH99] Ç.K. Koç and C. Paar, editors. *Cryptographic Hardware and Embedded Systems – CHES '99*. Springer-Verlag, LNCS 1717, 1999.
- [CH00] Ç.K. Koç and C. Paar, editors. *Cryptographic Hardware and Embedded Systems – CHES 2000*. Springer-Verlag, LNCS 1965, 2000.
- [CH01] Ç.K. Koç, D. Naccache and C. Paar, editors. *Cryptographic Hardware and Embedded Systems – CHES 2001*. Springer-Verlag, LNCS 2162, 2001.
- [CH02] B.S. Kaliski Jr., Ç.K. Koç and C. Paar, editors. *Cryptographic Hardware and Embedded Systems – CHES 2002*. Springer-Verlag, LNCS 2523, 2003.
- [CH03] C.D. Walter, Ç.K. Koç and C. Paar, editors. *Cryptographic Hardware and Embedded Systems – CHES 2003*. Springer-Verlag, LNCS 2779, 2003.
- [E90] I.B. Damgård, editor. *Advances in Cryptology – EUROCRYPT '90*. Springer-Verlag, LNCS 473, 1990.
- [E94] A. De Santis, editor. *Advances in Cryptology – EUROCRYPT '94*. Springer-Verlag, LNCS 950, 1994.
- [E97] W. Fumy, editor. *Advances in Cryptology – EUROCRYPT '97*. Springer-Verlag, LNCS 1233, 1997.
- [E00] B. Preneel, editor. *Advances in Cryptology – EUROCRYPT 2000*. Springer-Verlag, LNCS 1807, 2000.
- [E01] B. Pfitzmann, editor. *Advances in Cryptology – EUROCRYPT 2001*. Springer-Verlag, LNCS 2045, 2001.
- [E02] L. Knudsen, editor. *Advances in Cryptology – EUROCRYPT 2002*. Springer-Verlag, LNCS 2332, 2002.
- [E03] E. Biham, editor. *Advances in Cryptology – EUROCRYPT 2003*. Springer-Verlag, LNCS 2656, 2003.
- [E04] C. Cachin and J. Camenisch, editors. *Advances in Cryptology – EUROCRYPT 2004*. Springer-Verlag, LNCS 3027, 2003.
- [P01] K. Kim, editor. *Public Key Cryptography – PKC 2001*. Springer-Verlag, LNCS 1992, 2001.
- [P02] D. Naccache and P. Paillier, editors. *Public Key Cryptography – PKC 2002*. Springer-Verlag, LNCS 2274, 2002.
- [P03] Y.G. Desmedt, editor. *Public Key Cryptography – PKC 2003*. Springer-Verlag, LNCS 2567, 2003.
- [P04] F. Bao, editor. *Public Key Cryptography – PKC 2004*. Springer-Verlag, LNCS 2947, 2004.
- [ANSI X9.62] ANSI X9.62. *Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)*. American National Standards Institute, 1999.

- [ANSI X9.63] ANSI X9.63. *Public Key Cryptography for the Financial Services Industry: Elliptic Curve Key Agreement and Transport Protocols*. American National Standards Institute, 2001. Draft.
- [EU 1999] EU Directive 1999/93/EC of the European Parliament and of the Council. *On a Community Framework for Electronic Signatures*, December 1999.
- [FIPS 140.1] FIPS PUB 140-1. *Security requirements for cryptographic modules*. National Institute for Standards and Technology, 1994.
- [FIPS 180.1] FIPS PUB 180-1. *Secure Hash Standard*. National Institute for Standards and Technology, 1995.
- [FIPS 180.2] FIPS PUB 180-2. *Secure Hash Standard*. National Institute for Standards and Technology, 2001.
- [FIPS 186] FIPS PUB 186. *Digital Signature Standard (DSS)*. National Institute for Standards and Technology, 1994.
- [FIPS 186.2] FIPS PUB 186-2. *Digital Signature Standard (DSS)*. National Institute for Standards and Technology, 2000.
- [IBM CoPro] IBM Corporation. *IBM PCI Cryptographic Coprocessor—General Information Manual*, 6th ed., 2002.
- [IEEE 1363] IEEE 1363. *Standard Specifications for Public Key Cryptography*. IEEE, 2000.
- [ISO 15946-2] ISO X9.62. *International Standard 15946-2: Information Technology — Security Techniques — Cryptographic techniques based on elliptic curves — Part 2: Digital Signatures*. International Standards Organization, 2000.
- [NESSIE] NESSIE. *Security Evaluation Report*. NESSIE, 2002.
- [RFC 2412] IETF. *The Oakley Key Determination Protocol*, 1998.
- [RFC 3278] IETF. *The Use of Elliptic Curve Cryptography in the Cryptographic Message Syntax*, 2001.
- [SECG] SEC 1. *Elliptic Curve Cryptography*. Standards for Efficient Cryptography Group, 1999.
- [1] M. Abdalla, M. Bellare and P. Rogaway. DHAS: An encryption scheme based on the Diffie-Hellman problem. Submission to *P1363a: Standard Specifications for Public-Key Cryptography, Additional Techniques*, 2000.
- [2] L.M. Adleman. The function field sieve. In [A-1], 108–121.
- [3] L.M. Adleman and M.-D. Huang. Function field sieve method for discrete logarithms over finite fields. *Information and Computation*, **151**, 5–16, 1999.
- [4] L.M. Adleman, J. DeMarrais and M.-D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the jacobians of large genus hyperelliptic curves over finite fields. In [A-1], 28–40.
- [5] D. Agrawal, B. Archambeault, J.R. Rao and P. Rohatgi. The EM side-channel(s). In [CH02], 29–45.
- [6] T. Akishita and T. Takagi. Zero-value point attacks on elliptic curve cryptosystem. In C. Boyd and W. Mao, editors, *Information Security*, LNCS 2851, 218–233. Springer-Verlag, 2003.
- [7] M.-L. Akkar and C. Giraud. An implementation of DES and AES secure against some attacks. In [CH01], 309–318.
- [8] S.S. Al-Riyami and K.G. Paterson. Authenticated three party key agreement protocols from pairings. In K.G. Paterson, editor, *Cryptography and Coding*, LNCS 2898, 332–359. Springer-Verlag, 2003.
- [9] S.S. Al-Riyami and K.G. Paterson. Certificateless public key cryptography. In [A03], 452–473.
- [10] G. Appenzeller and B. Lynn. Minimal-overhead IP security using identity-based encryption. Preprint, 2003.

- [11] A.O.L. Atkin. The number of points on an elliptic curve modulo a prime. *Series of emails to the NMBRTHRY mailing list*, 1992.
- [12] R. Balasubramanian and N. Koblitz. The improbability that an elliptic curve has sub-exponential discrete log problem under the Menezes–Okamoto–Vanstone algorithm. *J. Cryptology*, **11**, 141–145, 1998.
- [13] D. Balfanz, G. Durfee, N. Shankar, D. Smetters, J. Staddon and H.-C. Wong. Secret handshakes from pairing-based key agreements. In *Proc. IEEE Symposium on Security and Privacy*, 180–196. IEEE, 2003.
- [14] R. Barua, R. Dutta and P. Sarkar. Extending Joux’s protocol to multi party key agreement. In T. Johansson and S. Maitra, editors, *Progress in Cryptology – INDOCRYPT 2003*, LNCS 2551, 205–217. Springer-Verlag, 2003.
- [15] P. Barreto. The pairing-based crypto lounge. <http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html>.
- [16] P.S.L.M. Barreto, H.Y. Kim, B. Lynn and M. Scott. Efficient algorithms for pairing-based cryptosystems. In [C02], 354–368.
- [17] P.S.L.M. Barreto, B. Lynn and M. Scott. Constructing elliptic curves with prescribed embedding degrees. In S. Cimato, C. Galdi and G. Persiano, editors, *Security in Communication Networks (SCN 2002)*, LNCS 2576, 257–267. Springer-Verlag, 2002.
- [18] P.S.L.M. Barreto, B. Lynn and M. Scott. On the selection of pairing-friendly groups. In M. Matsui and R. Zuccherato, editors, *Selected Areas in Cryptography – SAC 2003*, LNCS 3006, 17–25. Springer-Verlag, 2004.
- [19] M. Bellare, A. Desai, E. Jökipii and P. Rogaway. A concrete security treatment of symmetric encryption. In *Proc. of the 38th Symposium on Foundations of Computer Science*, IEEE, 1997.
- [20] M. Bellare, A. Desai, D. Pointcheval and P. Rogaway. Relations among notions of security for public-key encryption schemes. In [C98], 26–45.
- [21] M. Bellare, S. Goldwasser and D. Micciancio. “Pseudo-random” number generation within cryptographic algorithms: The DSS case. In [E97], 277–291.
- [22] M. Bellare, R. Canetti and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *Proc. of the 30th Annual Symposium on the Theory of Computing*, 419–428. ACM, 1998.
- [23] M. Bellare and A. Palacio. Protecting against key exposure: strongly key-insulated encryption with optimal threshold. See [EP], # 2002/064, 2002.
- [24] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proc. of the First ACM Conference on Computer and Communications Security*, 62–73. ACM, 1993.
- [25] M. Bellare and P. Rogaway. Entity authentication and key distribution. In [C93], 232–249.
- [26] I. Biehl, B. Meyer and V. Müller. Differential fault attacks on elliptic curve cryptosystems. In [C00], 131–146.
- [27] O. Billet and M. Joye. The Jacobi model of an elliptic curve and side-channel analysis. In M. Fossorier, T. Höholdt and A. Poli, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, LNCS 2643, 34–42. Springer-Verlag, 2003.
- [28] S. Blake-Wilson, D. Johnson and A.J. Menezes. Key agreement protocols and their security analysis. In *Cryptography and Coding*, LNCS 1355, 30–45. Springer-Verlag, 1997.
- [29] S. Blake-Wilson and A.J. Menezes. Security proofs for entity authentication and authenticated key transport protocols employing asymmetric techniques. In B. Christianson, B. Crispo, T. Lomas and M. Roe, editors, *Security Protocols*, LNCS 1361, 137–158. Springer-Verlag, 1997.

- [30] D. Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. In [C98], 1–12.
- [31] D. Bleichenbacher. On the generation of DSS one-time keys. Preprint, 2001.
- [32] A. Boldyreva. Efficient threshold signature, multisignature and blind signature schemes based on the gap-Diffie-Hellman-group signature scheme. In [P03], 31–46.
- [33] A. Boldyreva, A. Palacio and B. Warinschi. Secure proxy signature schemes for delegation of signing rights. See [EP], # 2003/096, 2003.
- [34] D. Boneh. The decision Diffie-Hellman problem. In [A-3], 48–63.
- [35] D. Boneh and X. Boyen. Short signatures without random oracles. In [E04], 56–73.
- [36] D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In [E04], 223–238.
- [37] D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano. Public key encryption with keyword search. In [E04], 506–522.
- [38] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In [C01], 213–229.
- [39] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. *SIAM J. Comp.*, **32**, 586–615, 2003.
- [40] D. Boneh, C. Gentry, B. Lynn and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In [E03], 416–432.
- [41] D. Boneh, A. Joux and P. Nguyen. Why textbook ElGamal and RSA encryption are insecure. In [A00], 30–43.
- [42] D. Boneh, B. Lynn and H. Shacham. Short signatures from the Weil pairing. In [A01], 514–532.
- [43] D. Boneh, B. Lynn and H. Shacham. Short signatures from the Weil pairing. Technical report, 2003. Revised version of [42], available from <http://crypto.stanford.edu/~dabo/abstracts/weilsigs.html>.
- [44] D. Boneh, I. Mironov and V. Shoup. Provably secure signature scheme from bilinear mapping. In M. Joye, editor, *Topics in Cryptology – CT-RSA 2003*, LNCS 2612, 98–110. Springer-Verlag, 2003.
- [45] I. Bouw, C. Diem and J. Scholten. Ordinary elliptic curves of high rank over $\mathbb{F}_p(x)$ with constant j -invariant. *Manuscripta Mathematica*, To appear.
- [46] W. Bosma, J. Cannon and C. Playoust. The Magma algebra system I: The user language. *J. Symbolic Comp.*, **24**, 3/4, 235–265, 1997.
- [47] C. Boyd, W. Mao and K.G. Paterson. Deniable authenticated key establishment for Internet protocols. In *Security Protocols*, LNCS. Springer-Verlag, To appear.
- [48] X. Boyen. Multipurpose identity-based signcryption: A swiss army knife for identity-based cryptography. In [C03], 382–398.
- [49] F. Brezing and A. Weng. Elliptic curves suitable for pairing based cryptography. See [EP], # 2003/143, 2003.
- [50] É. Brier, I. Déchène and M. Joye. Unified addition formulæ for elliptic curve cryptosystems. In N. Nedjah and L. de Macedo Mourelle, editors, *Embedded Cryptographic Hardware: Methodologies and Architectures*. Nova Science Publishers, 2004.
- [51] É. Brier and M. Joye. Weierstraß elliptic curves and side-channel attacks. In [P02], 335–345.
- [52] D.R.L. Brown. Generic groups, collision resistance and ECDSA. See [EP], # 2002/026, 2002.
- [53] M. Burmester and Y. Desmedt. A secure and efficient conference key distribution system. In [E94], 267–275.

- [54] R. Canetti, O. Goldreich and S. Halevi. The random oracle model, revisited. In *Proc. of the 30th Annual ACM Symposium on the Theory of Computing*, 209–218. ACM, 1998.
- [55] R. Canetti, S. Halevi and J. Katz. A forward-secure public-key encryption scheme. In [E03], 255–271.
- [56] R. Canetti, S. Halevi and J. Katz. Chosen-ciphertext security from identity-based encryption. See [E04], #, 2003.
- [57] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In [E01], 453–474.
- [58] R. Canetti, H. Krawczyk and J.B. Nielsen. Relaxing chosen-ciphertext security. In [C03], 565–582.
- [59] D.G. Cantor. Computing in the Jacobian of an hyperelliptic curve. *Math. Comp.*, **48**, 95–101, 1987.
- [60] R. Carls. A generalized arithmetic geometric mean (GAGM) sequence. Preprint, 2004.
- [61] J. Cathalo, F. Koeune and J.-J. Quisquater. A new type of timing attack: Application to GPS. In [CH03], 291–303.
- [62] J.C. Cha and J.H. Cheon. An identity-based signature from gap Diffie–Hellman groups. In [P03], 18–30.
- [63] L.S. Charlap and R. Coley. An elementary introduction to elliptic curves ii. Institute for Defense Analysis, CCR Expository Report 34, 1990.
- [64] S. Chari, C.S. Jutla, J.R. Rao and P. Rohatgi. Towards sound approaches to counteract power-analysis attacks. In [C99], 398–412.
- [65] D. Chaum. Security without identification: Transaction systems to make Big Brother obsolete. *Comm. ACM*, **28**, 1030–1044, 1985.
- [66] D. Chaum. Zero-knowledge undeniable signatures. In [E90], 458–464.
- [67] D. Chaum and H. van Antwerpen. Undeniable signatures. In [C89], 212–216.
- [68] L. Chen, K. Harrison, A. Moss, D. Soldera and N.P. Smart. Certification of public keys within an identity based system. In A.H. Chan and V.D. Gligor, editors, *Information Security*, LNCS 2433, 322–333. Springer-Verlag, 2002.
- [69] L. Chen, K. Harrison, D. Soldera and N.P. Smart. Applications of multiple trust authorities in pairing based cryptosystems. In G.I. Davida, Y. Frankel and O. Rees, editors, *Infrastructure Security, International Conference, InfraSec*, LNCS 2437, 260–275. Springer-Verlag, 2002.
- [70] L. Chen and C. Kudla. Identity based authenticated key agreement protocols from pairings. See [EP], # 2002/184, 2002.
- [71] L. Chen and C. Kudla. Identity based authenticated key agreement protocols from pairings. In *IEEE Computer Security Foundations Workshop*, 219–233. IEEE Computer Society Press, 2003.
- [72] L. Chen and J. Malone-Lee. Improved identity-based signcryption. Preprint, 2004.
- [73] X. Chen, F. Zhang and K. Kim. A new ID-based group signature scheme from bilinear pairings. See [EP], # 2003/116, 2003.
- [74] Z. Chen. Security analysis of Nalla-Reddy’s ID-based tripartite authenticated key agreement protocols. See [EP], # 2003/103, 2003.
- [75] J.H. Cheon. A universal forgery of Hess’s second ID-based signature against the known-message attack. See [EP], # 2002/028, 2002.
- [76] B. Chevallier-Mames, M. Ciet and M. Joye. Low-cost solutions for preventing simple side-channel analysis: Side-channel atomicity. *IEEE Trans. Computers*, **53**, 760–768, 2004.

- [77] D.V. Chudnovsky and G.V. Chudnovsky. Sequences of numbers generated by addition in formal groups and new primality and factorization tests. *Adv. Applied Math.*, **7**, 385–434, 1987.
- [78] M. Ciet and M. Joye. Elliptic curve cryptosystems in the presence of permanent and transient faults. *Designs, Codes and Cryptography*, To appear.
- [79] M. Ciet, J.-J. Quisquater and F. Sica. Preventing differential analysis in GLV elliptic curve scalar multiplication. In [CH02], 540–550.
- [80] M. Ciet, J.-J. Quisquater and F. Sica. A secure family of composite finite fields suitable for fast implementation of elliptic curve cryptography. In C. Pandu Rangan and C. Ding, editors, *Progress in Cryptology – INDOCRYPT 2001*, LNCS 2247, 108–116. Springer-Verlag, 2001.
- [81] C. Clavier, J.-S. Coron and N. Dabbous. Differential power analysis in the presence of hardware countermeasures. In [CH00], 252–263.
- [82] C. Clavier and M. Joye. Universal exponentiation algorithm: A first step towards provable SPA-resistance. In [CH01], 300–308.
- [83] C. Cocks. An identity based encryption scheme based on quadratic residues. In B. Honary, editor, *Cryptography and Coding*, LNCS 2260, 360–363. Springer-Verlag, 2001.
- [84] C. Cocks and R.G.E. Pinch. ID-based cryptosystems based on the Weil pairing. Unpublished manuscript, 2001.
- [85] H. Cohen, A. Miyaji and T. Ono. Efficient elliptic curve exponentiation using mixed coordinates. In [A98], 51–65.
- [86] D. Coppersmith. Fast evaluation of logarithms in fields of characteristic 2. *IEEE Trans. Inf. Theory*, **30**, 587–594, 1984.
- [87] J.-S. Coron. Resistance against differential power analysis for elliptic curve cryptosystems. In [CH99], 292–302.
- [88] J.-S. Coron and L. Goubin. On Boolean and arithmetic masking against differential power analysis. In [CH00], 231–237.
- [89] J.-S. Coron and D. Naccache. Boneh *et al.*'s k -element aggregate extraction assumption is equivalent to the Diffie-Hellman assumption. In [A03], 392–397.
- [90] J.-M. Couveignes. *Quelques calculs en théorie des nombres*. PhD thesis, Université de Bordeaux, 1994.
- [91] J.-M. Couveignes. Computing l -isogenies with the p -torsion. In [A-2], 59–65.
- [92] J.-M. Couveignes. Algebraic groups and discrete logarithms. In *Public Key Cryptography and Computational Number Theory*, 17–27. Walter de Gruyter, 2001.
- [93] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM Journal on Computing*, **33**(1), 167–226, 2004.
- [94] C.R. Dalton. The NHS as a proving ground for cryptosystems. *Information Security Technical Report*, **8**(3), 73–88, 2003.
- [95] B. den Boer, K. Lemke and G. Wicke. A DPA attack against the modular reduction within a CRT implementation of RSA. In [CH02], 228–234.
- [96] J. Denef and F. Vercauteren. An extension of Kedlaya's algorithm to Artin-Schreier curves in characteristic 2. In [A-5], 308–323.
- [97] J. Denef and F. Vercauteren. An extension of Kedlaya's algorithm to hyperelliptic curves in characteristic 2. *J. Cryptology*, To appear.
- [98] J. Denef and F. Vercauteren. Computing zeta functions of C_{ab} curves using Monsky-Washnitzer cohomology. Preprint, 2003.
- [99] A.W. Dent. An evaluation of EPOC-2. NESSIE, Public report, 2001.
- [100] A.W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In [A02], 100–109.

- [101] M. Deuring. Die Typen der Multiplikatorringe elliptischer Funktionenkörper. *Abh. Math. Sem. Univ. Hamburg*, **14**, 197–272, 1941.
- [102] E. De Win, S. Mister, B. Preneel and M. Wiener. On the performance of signature schemes based on elliptic curves. In [A-3], 252–266.
- [103] C. Diem. *A study on theoretical and practical aspects of Weil-restrictions of varieties*. PhD thesis, Universität-Gesamthochschule Essen, 2001.
- [104] C. Diem. The GHS-attack in odd characteristic. *J. Ramanujan Math. Soc.*, **18**, 2002.
- [105] C. Diem and N. Naumann. On the structure of Weil restrictions of abelian varieties. *J. Ramanujan Math. Soc.*, **18**, 2003.
- [106] C. Diem and J. Scholten. Cover attacks – a report for the AREHCC project. Preprint, 2003.
- [107] Y. Dodis, M. Franklin, J. Katz, A. Miyaji and M. Yung. Intrusion-resilient public-key encryption. In M. Joye, editor, *Topics in Cryptology – CT-RSA 2003*, LNCS 2612, 19–32. Springer-Verlag, 2003.
- [108] R. Dupont and A. Enge. Practical non-interactive key distribution based on pairings. *Discrete Applied Mathematics*, To appear.
- [109] R. Dupont, A. Enge and F. Morain. Building curves with arbitrary small MOV degree over finite prime fields. See [EP], # 2002/094, 2002.
- [110] I.M. Duursma. Class numbers for some hyperelliptic curves. In R. Pellikaan, M. Perret and S.G. Vladut, editors, *Arithmetic, Geometry and Coding Theory*, 45–52. Walter de Gruyter, 1996.
- [111] I.M. Duursma, P. Gaudry and F. Morain. Speeding up the discrete log computation on curves with automorphisms. In [A99], 103–121.
- [112] I.M. Duursma and H.-S. Lee. Tate pairing implementation for hyperelliptic curves. In [A03], 111–222.
- [113] B. Dwork. On the rationality of the zeta function of an algebraic variety. *Amer. J. Math.*, **82**, 631–648, 1960.
- [114] K. Eisenträger, K. Lauter and P.L. Montgomery. Fast elliptic curve arithmetic and improved Weil pairing evaluation. In M. Joye, editor, *Topics in Cryptology – CT-RSA 2003*, LNCS 2612, 343–354. Springer-Verlag, 2003.
- [115] N. Elkies. Elliptic and modular curves over finite fields and related computational issues. In *Computational Perspectives on Number Theory*, 21–76, 1998.
- [116] A. Enge. Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time. *Math. Comp.*, **71**, 729–742, 2002.
- [117] A. Enge and P. Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta Arith.*, **102**, 83–103, 2002.
- [118] A. Enge and A. Stein. Smooth ideals in hyperelliptic function fields. *Math. Comp.*, **71**, 1219–1230, 2002.
- [119] P. Fahn and P. Pearson. IPA: A new class of power attacks. In [CH99], 173–186.
- [120] W. Fischer, C. Giraud, E.W. Knudsen and J.-P. Seifert. Parallel scalar multiplication on general elliptic curves over $\mathbb{F}_p$ hedged against non-differential side-channel attacks. See [EP], # 2002/007, 2002.
- [121] R. Flassenberg and S. Paulus. Sieving in function fields. *Experiment. Math.*, **8**, 339–349, 1999.
- [122] P.-A. Fouque and F. Valette. The doubling attack – Why upwards is better than downwards. In [CH03], 269–280.
- [123] M. Fouquet, P. Gaudry and R. Harley. On Satoh’s algorithm and its implementation. *J. Ramanujan Math. Soc.*, **15**, 281–318, 2000.
- [124] G. Frey. How to disguise an elliptic curve. Talk at ECC’ 98, Waterloo, 1998.

- [125] G. Frey, M. Müller and H.-G. Rück. The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems. *IEEE Trans. Inf. Theory*, **45**, 1717–1719, 1999.
- [126] G. Frey and H.-G. Rück. A remark concerning m -divisibility and the discrete logarithm problem in the divisor class group of curves. *Math. Comp.*, **62**, 865–874, 1994.
- [127] G. Frey. Applications of arithmetical geometry to cryptographic constructions. In D. Jungnickel and H. Niederreiter, editors, *Finite Fields and Applications – 5*, 128–161. Springer, 2001.
- [128] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In [C99], 537–554.
- [129] S.D. Galbraith. Constructing isogenies between elliptic curves over finite fields. *LMS J. Comput. and Math.*, **2**, 118–138, 1999.
- [130] S.D. Galbraith. Supersingular curves in cryptography. In [A01], 495–513.
- [131] S.D. Galbraith. Weil descent of Jacobians. In D. Augot and C. Carlet, editors, *WCC2001 International workshop on coding and cryptography*, Electron. Notes Discrete Math. 6. Elsevier, 2001.
- [132] S.D. Galbraith. Limitations of constructive Weil descent. In *Public Key Cryptography and Computational Number Theory*, 59–70. Walter de Gruyter, 2001.
- [133] S.D. Galbraith, K. Harrison and D. Soldera. Implementing the Tate pairing. In [A-5], 324–337.
- [134] S.D. Galbraith, F. Hess and N.P. Smart. Extending the GHS Weil descent attack. In [E02], 29–44.
- [135] S.D. Galbraith, H.J. Hopkins and I.E. Shparlinski. Secure Bilinear Diffie-Hellman bits. In J. Pieprzyk H. Wang and V. Varadharajan, editors, *Information Security and Privacy – ACISP 2004*, LNCS 3108, 370–378. Springer-Verlag, 2004.
- [136] S.D. Galbraith and J. McKee. The probability that the number of points on an elliptic curve over a finite field is prime. *J. London Math. Soc.*, **62**, 671–684, 2000.
- [137] S.D. Galbraith and N.P. Smart. A cryptographic application of Weil descent. In M. Walker, editor, *Cryptography and Coding*, LNCS 1746, 191–200. Springer-Verlag, 1999.
- [138] R. Gallant, R. Lambert and S. Vanstone. Improving the parallelized Pollard lambda search on binary anomalous curves. *Math. Comp.*, **69**, 1699–1705, 2000.
- [139] K. Gandolfi, C. Mourtel and F. Olivier. Electromagnetic analysis: Concrete results. In [CH01], 251–261.
- [140] T. Garefalakis. The generalised Weil pairing and the discrete logarithm problem on elliptic curves. In S. Rajsbaum, editor, *LATIN 2002: Theoretical Informatics*, LNCS 2286, 118–130. Springer-Verlag, 2002.
- [141] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.
- [142] P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In [E00], 19–34.
- [143] P. Gaudry. A comparison and a combination of SST and AGM algorithms for counting points of elliptic curves in characteristic 2. In [A02], 311–327.
- [144] P. Gaudry and N. Gürel. An extension of Kedlaya’s point-counting algorithm to superelliptic curves. In [A01], 480–494.
- [145] P. Gaudry, F. Hess and N.P. Smart. Constructive and destructive facets of Weil descent on elliptic curves. *J. Cryptology*, **15**, 19–46, 2002.
- [146] C. Gentry. Certificate-based encryption and the certificate revocation problem. In [E03], 272–293.

- [147] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography. In [A02], 548–566.
- [148] R. Gerkmann. *The p -adic cohomology of varieties over finite fields and applications to the computation of zeta functions*. PhD thesis, Universität Duisberg-Essen, 2003.
- [149] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comp. Syst. Sci.*, **28**, 270–299, 1984.
- [150] S. Goldwasser, S. Micali and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Comp.*, **17**, 281–308, 1988.
- [151] L. Goubin. A refined power-analysis attack on elliptic curve cryptosystems. In [P03], 199–210.
- [152] L. Goubin and J. Patarin. DES and differential power analysis – The duplication method. In [CH99], 158–172.
- [153] L. Granboulan. RSA hybrid encryption schemes. Available from <http://www.di.ens.fr/~granboul/recherche/publications/abs-2001-RSAenc.html>, 2001.
- [154] R. Granger. Estimates for discrete logarithm computations in finite fields. In K.G. Paterson, editor, *Cryptography and Coding*, LNCS 2898, 190–206. Springer-Verlag, 2003.
- [155] G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 2nd ed., 1992.
- [156] J. Ha and S. Moon. Randomized signed-scalar multiplication of ECC to resist power attacks. In [CH02], 551–563.
- [157] H. Handschuh, P. Paillier and J. Stern. Probing attacks on tamper-resistant devices. In [CH99], 303–315.
- [158] R. Harley. Asymptotically optimal p -adic point-counting. Email to normal-font@NMBRTHRY mailing list, December 2002.
- [159] R. Harley. Method for solving Frobenius equations for elliptic-curve cryptography. United States Patent Application, December 2003.
- [160] R. Harley and J.-F. Mestre. Method for generating secure elliptic curves using an arithmetic-geometric mean iteration. United States Patent Application, June 2002.
- [161] M.A. Hasan. Power analysis attacks and algorithmic approaches to their countermeasures for Koblitz cryptosystems. In [CH00], 93–108.
- [162] J. Herranz and G. Sáez. A provably secure ID-based ring signature scheme. See [EP], # 2003/261, 2003.
- [163] F. Hess. Exponent group signature schemes and efficient identity based signature schemes based on pairings. See [EP], # 2002/012, 2002.
- [164] F. Hess. Efficient identity based signature schemes based on pairings. In K. Nyberg and H. Heys, editors, *Selected Areas in Cryptography – SAC 2002*, LNCS 2595, 310–324. Springer-Verlag, 2003.
- [165] F. Hess. The GHS attack revisited. In [E03], 374–387.
- [166] F. Hess. On the security of the verifiably-encrypted signature scheme of Boneh, Gentry, Lynn and Shacham. *Info. Proc. Lett.*, **89**, 111–114, 2004.
- [167] F. Hess. A note on the Tate pairing of curves over finite fields. *Arch. Math.*, **82**, 28–32, 2004.
- [168] F. Hess. Generalising the GHS attack on the elliptic curve discrete logarithm problem. *LMS J. Comput. and Math.*, **7**, 167–192, 2004.
- [169] F. Hess. Computing relations in divisor class groups of algebraic curves over finite fields. Preprint, 2003.
- [170] F. Hess, N. P. Smart and G. Seroussi. Two topics in hyperelliptic cryptography. In S. Vaudenay and A.M. Youssef, editors, *Selected Areas in Cryptography – SAC 2001*, LNCS 2259, 181–189. Springer-Verlag, 2001.

- [171] Y. Hitchcock and P. Montague. A new elliptic curve scalar multiplication algorithm to resist simple power analysis. In L.M. Batten and J. Seberry, editors, *Information Security and Privacy – ACISP 2002*, LNCS 2384, 214–225. Springer-Verlag, 2002.
- [172] J. Horowitz and B. Lynn. Toward hierarchical identity-based encryption. In [E02], 466–481.
- [173] E. W. Howe. The Weil pairing and the Hilbert symbol. *Math. Ann.*, **305**, 387–392, 1996.
- [174] N. Howgrave-Graham and N.P. Smart. Lattice attacks on digital signature schemes. *Designs, Codes and Cryptography*, **23**, 283–290, 2001.
- [175] K. Itoh, T. Izu and M. Takaneke. Address-bit differential power analysis of cryptographic schemes OK-ECDH and OK-ECDSA. In [CH02], 129–143.
- [176] K. Itoh, T. Izu and M. Takaneke. A practical countermeasure against address-bit differential power analysis. In [CH03], 382–396.
- [177] K. Itoh, J. Yajima, M. Takaneke and N. Torii. DPA countermeasures by improving the window method. In [CH02], 303–317.
- [178] T. Izu and T. Takagi. A fast parallel elliptic curve multiplication resistant against side channel attacks. In [P02], 280–296.
- [179] T. Izu and T. Takagi. Exceptional procedure attack on elliptic curve cryptosystems. In [P03], 224–239.
- [180] T. Izu and T. Takagi. Efficient computations of the Tate pairing for the large MOV degrees. In P.J. Lee and C.H. Lim, editors, *Information Security and Cryptology – ICISC 2002*, LNCS 2587, 283–297. Springer-Verlag, 2003.
- [181] M. Jacobson, A.J. Menezes and A. Stein. Solving elliptic curve discrete logarithm problems using Weil descent. *J. Ramanujan Math. Soc.*, **16**, 231–260, 2001.
- [182] M. Jacobson and A. van der Poorten. Computational aspects of NUCOMP. In [A-5], 120–133.
- [183] A. Joux. A one round protocol for tripartite Diffie–Hellman. In [A-4], 385–394.
- [184] A. Joux. The Weil and Tate pairings as building blocks for public key cryptosystems. In [A-5], 20–32.
- [185] A. Joux and R. Lercier. The function field sieve is quite special. In [A-5], 431–445.
- [186] A. Joux and K. Nguyen. Separating Decision Diffie–Hellman from Diffie–Hellman in cryptographic groups. *J. Cryptology*, **16**, 239–248, 2003.
- [187] M. Joye. Recovering lost efficiency of exponentiation algorithms on smart cards. *Elect. Lett.*, **38**, 1095–1097, 2002.
- [188] M. Joye and J.-J. Quisquater. Hessian elliptic curves and side-channel attacks. In [CH01], 402–410.
- [189] M. Joye, J.-J. Quisquater and M. Yung. On the power of misbehaving adversaries and security analysis of the original EPOC. In D. Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, LNCS 2020, 208–222. Springer-Verlag, 2001.
- [190] M. Joye and C. Tymen. Protections against differential analysis for elliptic curve cryptography: An algebraic approach. In [CH01], 377–390.
- [191] M. Joye and S.-M. Yen. The Montgomery powering ladder. In [CH02], 291–302.
- [192] B.G. Kang, J.H. Park and S.G. Hahn. A certificate-based signature scheme. In T. Okamoto, editor, *Topics in Cryptology – CT-RSA 2004*, LNCS 2964, 99–111. Springer-Verlag, 2004.
- [193] Kant group. *Kash*. <http://www.math.tu-berlin.de/~kant>, 2003.
- [194] A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers on automata. *Soviet Physics Doklady*, **7**, 595–596, 1963.
- [195] C. Karlof and D. Wagner. Hidden Markov model cryptanalysis. In [CH03], 17–34.
- [196] K.S. Kedlaya. Counting points on hyperelliptic curves using Monsky-Washnitzer cohomology. *J. Ramanujan Math. Soc.*, **16**, 323–338, 2001.

- [197] J. Kempf, C. Gentry and A. Silverberg. Securing IPv6 neighbor discovery using address based keys (ABKS). Internet Draft Document, expired December 2002, 2002. Available from <http://www.docomolabs-usa.com/pdf/PS2003-080.pdf>.
- [198] A. Khalili, J. Katz and W.A. Arbaugh. Toward secure key distribution in truly ad-hoc networks. In *Proc. 2003 Symposium on Applications and the Internet Workshops (SAINT 2003)*. IEEE Computer Society, 2003.
- [199] H.Y. Kim, J.Y. Park, J.H. Cheon, J.H. Park, J.H. Kim. and S.G. Hahn. Fast elliptic curve point counting using Gaussian Normal Basis. In [A-5], 292–307.
- [200] V. Klima and T. Rosa. Further results and considerations on side channel attacks on RSA. In [CH02], 244–259.
- [201] N. Koblitz. *p-Adic Numbers, p-Adic Analysis, and Zeta-Functions*. Springer-Verlag, GTM 58, 1984.
- [202] N. Koblitz. CM curves with good cryptographic properties. In [C91], 279–287.
- [203] N. Koblitz. *Algebraic Aspects of Cryptography*. Springer-Verlag, 1997.
- [204] H. Koch. *Algebraic Number Theory*. Springer-Verlag, 2nd ed., 1997.
- [205] P.C. Kocher. Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems. In [C96], 104–113.
- [206] P.C. Kocher, J. Jaffe and B. Jun. Differential power analysis. In [C99], 388–397.
- [207] D.R. Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California, Berkeley, 1996.
- [208] D.R. Kohel. The AGM- $X_0(N)$ Heegner point lifting algorithm and elliptic curve point counting. In [A03], 124–136.
- [209] D.R. Kohel and I.E. Shparlinski. On exponential sums and group generators for elliptic curves over finite fields. In [A-4], 395–404.
- [210] S. Lang. *Algebra*. Addison-Wesley, 3rd ed., 1993.
- [211] T. Lange. *Efficient arithmetic on hyperelliptic curves*. PhD thesis, Universität-Gesamthochschule Essen, 2001.
- [212] T. Lange. Formulae for arithmetic on genus 2 hyperelliptic curves. *Appl. Algebra Engrg. Comm. Comput.*, To appear.
- [213] A.G.B. Lauder and D. Wan. Counting points on varieties over finite fields of small characteristic. In J.P. Buhler and P. Stevenhagen, editors, *Algorithmic Number Theory: Lattices, Number Fields, Curves and Cryptography*. Mathematical Sciences Research Institute Publications, 2002. To appear.
- [214] A.G.B. Lauder and D. Wan. Computing zeta functions of Artin-Schreier curves over finite fields. *LMS J. Comput. and Math.*, **5**, 34–55, 2002.
- [215] A.G.B. Lauder and D. Wan. Computing zeta functions of Artin-Schreier curves over finite fields II. *J. Complexity*, **20**, 331–349, 2004.
- [216] L. Law, A.J. Menezes, M. Qu, J. Solinas and S. Vanstone. An efficient protocol for authenticated key agreement. *Designs, Codes and Cryptography*, **28**, 119–134, 2003.
- [217] R. Lercier. *Algorithmique des courbes elliptiques dans les corps finis*. PhD thesis, École Polytechnique, 1997.
- [218] R. Lercier and D. Lubicz. Counting points on elliptic curves over finite fields of small characteristic in quasi quadratic time. In [E03], 360–373.
- [219] R. Lercier and D. Lubicz. A quasi quadratic time algorithm for hyperelliptic curve point counting. Preprint, 2003.
- [220] P.-Y. Liardet and N.P. Smart. Preventing SPA/DPA in ECC systems using the Jacobi form. In [CH01], 391–401.
- [221] B. Libert and J.-J. Quisquater. New identity based signcryption schemes from pairings. See [EP], # 2003/023, 2003.

- [222] B. Libert and J.-J. Quisquater. Identity based undeniable signatures. In T. Okamoto, editor, *Topics in Cryptology – CT-RSA 2004*, LNCS 2964, 112–125. Springer-Verlag, 2004.
- [223] B. Libert and J.-J. Quisquater. Efficient signcryption with key privacy from gap Diffie-Hellman groups. In [P04], 187–200.
- [224] S. Lichtenbaum. Duality theorems for curves over p -adic fields. *Inventiones Math.*, **7**, 120–136, 1969.
- [225] J. López and R. Dahab. Fast multiplication on elliptic curves over $GF(2^m)$ without precomputation. In [CH99], 316–327.
- [226] D. Lorenzini. *An Invitation to Arithmetic Geometry*. AMS, Graduate Studies in Mathematics 106, 1993.
- [227] J. Lubin, J.-P. Serre and J. Tate. Elliptic curves and formal groups. *Lecture notes prepared in connection with the seminars held at the Summer Institute on Algebraic Geometry*, Whitney Estate, Woods Hole, Massachusetts, 1964.
- [228] B. Lynn. Authenticated identity-based encryption. See [EP], # 2002/072, 2002.
- [229] Magma Comp. algebra group. *Magma*. Available from <http://www.maths.usyd.edu.au:8000/u/magma/>, 2003.
- [230] J. Malone-Lee. Identity-based signcryption. See [EP], # 2002/098, 2002.
- [231] J. Malone-Lee. Signcryption with non-interactive non-repudiation, To appear.
- [232] J. Manger. A chosen ciphertext attack on RSA optimal asymmetric encryption padding (OAEP) as standardized in PKCS # 1 v2.0. In [C01], 230–238.
- [233] M. Maurer, A.J. Menezes and E. Teske. Analysis of the GHS Weil descent attack on the ECDLP over characteristic two finite fields of composite degree. *LMS J. Comput. and Math.*, **5**, 127–174, 2002.
- [234] D. May, H.L. Muller and N.P. Smart. Random register renaming to foil DPA. In [CH01], 28–38.
- [235] R. Mayer-Sommer. Smartly analyzing the simplicity and the power of simple power analysis on smartcards. In [CH00], 78–92.
- [236] A. Miyaji, T. Ono and H. Cohen. Efficient elliptic curve exponentiation. In Y. Han, T. Okamoto and S. Qing, editors, *Information and Communications Security (ICICS '97)*, LNCS 1334, 282–290. Springer-Verlag, 1997.
- [237] W. Meier and O. Staffelbach. Efficient multiplication on certain non-supersingular elliptic curves. In [C92], 333–344.
- [238] A.J. Menezes. *Elliptic Curve Public Key Cryptosystems*. Kluwer, 1993.
- [239] A.J. Menezes, T. Okamoto and S.A. Vanstone. Reducing elliptic curve logarithms to a finite field. *IEEE Trans. Inf. Theory*, **39**, 1639–1646, 1993.
- [240] A.J. Menezes, P.C. van Oorschot and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
- [241] A.J. Menezes and M. Qu. Analysis of the Weil descent attack of Gaudry, Hess and Smart. In D. Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, LNCS 2020, 308–318. Springer-Verlag, 2001.
- [242] A.J. Menezes, E. Teske and A. Weng. Weak fields for ECC. In T. Okamoto, editor, *Topics in Cryptology – CT-RSA 2004*, LNCS 2964, 366–386. Springer-Verlag, 2004.
- [243] A.J. Menezes, Y.-H. Wu and R. Zuccherato. An elementary introduction to hyper-elliptic curves. In [203], 155–178.
- [244] T.S. Messerges. Using second-order power analysis to attack DPA resistant software. In [CH00], 238–251.
- [245] T.S. Messerges, E.A. Dabbish and R.H. Sloan. Power analysis attacks of modular exponentiation in smartcards. In [CH99], 144–157.
- [246] W. Messing. *The crystals associated to Barsotti-Tate groups: with applications to abelian schemes*. Springer-Verlag, GTM 264, 1972.

- [247] J.-F. Mestre. Lettre adressée à Gaudry et Harley, December 2000. Available at <http://www.math.jussieu.fr/~mestre/>.
- [248] J.-F. Mestre. Algorithmes pour compter des points de courbes en petite caractéristique et en petit genre, March 2002. Available at <http://www.math.jussieu.fr/~mestre/>.
- [249] V. Miller. Short programs for functions on curves. Unpublished manuscript, 1986.
- [250] S. Mitsunari, R. Sakai and M. Kasahara. A new traitor tracing schemes using bilinear map. *IEICE Trans. Fundamentals*, **E84**, 481–484, 2002.
- [251] A. Miyaji, M. Nakabayashi and S. Takano. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Trans. Fundamentals*, **E84**, 1234–1243, 2001.
- [252] R.T. Moenck. Fast computation of GCDs. In *Proc. 5th Annual ACM Symposium on the Theory of Computing*, 142–151. ACM, 1973.
- [253] B. Möller. Securing elliptic curve point multiplication against side-channel attacks. In G.I. Davida and Y. Frankel, editors, *Information Security*, LNCS 2200, 324–334. Springer-Verlag, 2001.
- [254] P.L. Montgomery. Modular multiplication without trial division. *Math. Comp.*, **44**, 519–521, 1985.
- [255] P.L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. *Math. Comp.*, **48**, 243–264, 1987.
- [256] F. Morain and J. Olivos. Speeding up the computations on an elliptic curve using addition-subtraction chains. *Theoretical Informatics and Applications*, **24**, 531–543, 1990.
- [257] V. Müller, A. Stein and C. Thiel. Computing discrete logarithms in real quadratic congruence function fields of large genus. *Math. Comp.*, **68**, 807–822, 1999.
- [258] K. Nagao. Improving group law algorithms for Jacobians of hyperelliptic curves. In **[A-4]**, 439–447.
- [259] D. Nalla and K.C. Reddy. ID-based tripartite authenticated key agreement protocols from pairings. See **[EP]**, # 2003/04, 2003.
- [260] J. Neukirch. *Algebraic Number Theory*. Springer-Verlag, 1999.
- [261] P.Q. Nguyen and I.E. Shparlinski. The insecurity of the Digital Signature Algorithm with partially known nonces. *J. Cryptology*, **15**, 151–176, 2002.
- [262] P.Q. Nguyen and I.E. Shparlinski. The insecurity of the Elliptic Curve Digital Signature Algorithm with partially known nonces. *Designs, Codes and Cryptography*, **30**, 201–217, 2003.
- [263] T. Okamoto and D. Pointcheval. The gap problems: A new class of problems for the security of cryptographic schemes. In **[P01]**, 104–118.
- [264] K. Okeya and K. Sakurai. Power analysis breaks elliptic curve cryptosystems even secure against the timing attack. In B. Roy and E. Okamoto, editors, *Progress in Cryptology – INDOCRYPT 2000*, LNCS 1977, 178–190. Springer-Verlag, 2000.
- [265] K. Okeya and K. Sakurai. On insecurity of the side channel attack countermeasure using addition-subtraction chains under distinguishability between addition and doubling. In L. Batten and J. Seberry, editors, *Information Security and Privacy – ACISP 2002*, LNCS 2384, 420–435. Springer-Verlag, 2002.
- [266] P.C. van Oorschot and M.J. Wiener. Parallel collision search with cryptanalytic applications. *J. Cryptology*, **12**, 1–28, 1999.
- [267] G. Orlando and C. Paar. A high performance reconfigurable elliptic curve processor for $GF(2^m)$. In **[CH00]**, 41–56.
- [268] S.B. Örs, E. Oswald and B. Preneel. Power-analysis attacks on FPGAs – First experimental results. In **[CH03]**, 35–50.
- [269] E. Oswald. Enhancing simple power-analysis attacks on elliptic curve cryptosystems. In **[CH02]**, 82–97.

- [270] E. Oswald. Markov model side-channel analysis. Unpublished manuscript, 2003.
- [271] E. Oswald and M. Aigner. Randomized addition-subtraction chains as a counter-measure against power attacks. In [CH01], 39–50.
- [272] K.G. Paterson. ID-based signatures from pairings on elliptic curves. *Elect. Lett.*, **38**, 1025–1026, 2002.
- [273] K.G. Paterson. Cryptography from pairings: a snapshot of current research. *Information Security Technical Report*, **7**, 41–54, 2002.
- [274] K.G. Paterson and G. Price. A comparion between traditional PKIs and identity-based cryptography. *Information Security Technical Report*, **8**, 57–72, 2003.
- [275] S. Paulus and A. Stein. Comparing real and imaginary arithmetics for divisor class groups of hyperelliptic curves. In [A-3], 576–591.
- [276] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *J. Cryptology*, **13**, 361–396, 2000.
- [277] J. Pelzl, T. Wollinger, J. Guajardo and C. Paar. Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves. In [CH03], 351–365.
- [278] J.-J. Quisquater and D. Samyde. Electromagnetic analysis (EMA): Measures and counter-measures for smart cards. In S. Attali and T. Jensen, editors, *Smart Card Programming and Security (E-smart 2001)*, LNCS 2140, 200–210. Springer-Verlag, 2001.
- [279] M.O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. MIT Laboratory for Computer Science, Technical Report MIT/LCS/TR-212, 1979.
- [280] C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In [C91], 434–444.
- [281] K.C. Reddy and D. Nalla. Identity based authenticated group key agreement protocol. In A.J. Menezes and P. Sarkar, editors, *Progress in Cryptology – INDOCRYPT 2002*, LNCS 2551, 215–233. Springer-Verlag, 2002.
- [282] C. Ritzenthaler. Point counting on genus 3 non hyperelliptic curves. In [A-6], 379–394.
- [283] H.G. Rück. On the discrete logarithm in the divisor class group of curves. *Math. Comp.*, **68**, 805–806, 1999.
- [284] R. Sakai, K. Ohgishi and M. Kasahara. Cryptosystems based on pairing. In *2000 Symposium on Cryptography and Information Security – SCIS 2000*, 2000.
- [285] T. Satoh. The canonical lift of an ordinary elliptic curve over a finite field and its point counting. *J. Ramanujan Math. Soc.*, **15**, 247–270, 2000.
- [286] T. Satoh. On p -adic point counting algorithms for elliptic curves over finite fields. In [A-5], 43–66.
- [287] T. Satoh, B. Skjærnaa and Y. Taguchi. Fast computation of canonical lifts of elliptic curves and its application to point counting. *Finite Fields Appl.*, **9**, 89–101, 2003.
- [288] W. Schindler. A timing attack against RSA with the Chinese remainder theorem. In [CH00], 109–124.
- [289] W. Schindler. A combined timing and power attack. In [P02], 263–279.
- [290] J. Scholten. Weil restriction of an elliptic curve over a quadratic extension. Preprint, 2004.
- [291] A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing (Arch. Elektron. Rechnen)*, **7**, 281–292, 1971.
- [292] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p . *Math. Comp.*, **44**, 483–494, 1985.
- [293] R. Schoof. Nonsingular plane cubic curves over finite fields. *J. Combin. Theory Ser. A*, **46**, 183–211, 1987.

- [294] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, **7**, 219–254, 1995.
- [295] J.-P. Serre. *Local Fields*. Springer-Verlag, GTM 67, 1979.
- [296] A. Shamir. Identity based cryptosystems and signature schemes. In [C84], 47–53.
- [297] A. Shamir. Protecting smart cards from passive power analysis with detached power supplies. In [CH00], 71–77.
- [298] D. Shanks. On Gauss and composition I and II. In R. Mollin, editor, *Number Theory and its Applications*, 163–204. Kluwer Academic Publishers, 1989.
- [299] K. Shim. A man-in-the-middle attack on Nalla-Reddy’s ID-based tripartite authenticated key agreement protocol. See [EP], # 2003/115, 2003.
- [300] K. Shim. Cryptanalysis of Al-Riyami-Paterson’s authenticated three party key agreement protocols. See [EP], # 2003/122, 2003.
- [301] K. Shim. Efficient one round tripartite authenticated key agreement protocol from Weil pairing. *Elect. Lett.*, **39**, 208–209, 2003.
- [302] K. Shim. Efficient ID-based authenticated key agreement protocol based on Weil pairing. *Elect. Lett.*, **39**, 653–654, 2003.
- [303] V. Shoup. Lower bounds for discrete logarithms and related problems. In [C97], 256–266.
- [304] V. Shoup. Using hash functions as a hedge against chosen ciphertext attack. In [E00], 275–288.
- [305] V. Shoup. A proposal for an ISO standard for public key encryption, v2.1. Preprint, 2001.
- [306] A. Silverberg and K. Rubin. Supersingular abelian varieties in cryptology. In [C02], 336–353.
- [307] J.H. Silverman. *The Arithmetic of Elliptic Curves*. Springer-Verlag, GTM 106, 1986.
- [308] B. Skjærnaa. Satoh’s algorithm in characteristic 2. *Math. Comp.*, **72**, 477–487, 2003.
- [309] N.P. Smart. The Hessian form of an elliptic curve. In [CH01], 118–125.
- [310] N.P. Smart. The exact security of ECIES in the generic group model. In B. Honary, editor, *Coding and Cryptography*, LNCS 2260, 73–84. Springer-Verlag, 2001.
- [311] N.P. Smart. How secure are elliptic curves over composite extension fields? In [E01], 30–39.
- [312] N.P. Smart. An identity based authenticated key agreement protocol based on the Weil pairing. *Elect. Lett.*, **38**, 630–632, 2002.
- [313] N.P. Smart. An analysis of Goubin’s refined power analysis attack. In [CH03], 281–290.
- [314] N.P. Smart. Access control using pairing based cryptography. In M. Joye, editor, *Topics in Cryptology – CT-RSA 2003*, LNCS 2612, 111–121. Springer-Verlag, 2003.
- [315] D.K. Smetters and G. Durfee. Domain-based administration of identity-based cryptosystems for secure email and IPSEC. In *Proc. 12th USENIX Security Symposium*, 215–229, 2003.
- [316] J. Solinas. Efficient arithmetic on Koblitz curves. *Designs, Codes and Cryptography*, **19**, 195–249, 2000.
- [317] A. Stein. Sharp upper bounds for arithmetics in hyperelliptic function fields. *J. Ramanujan Math. Soc.*, **16**, 1–86, 2001.
- [318] E. Steinfeld, L. Bull, H. Wang and J. Pieprzyk. Universal designated-verifier signatures. In [A03], 523–542.
- [319] H. Stichtenoth. *Algebraic Function Fields and Codes*. Springer-Verlag, 1993.
- [320] H. Stichtenoth and C. Xing. On the structure of the divisor class group of a class of curves over finite fields. *Arch. Math.*, **65**, 141–150, 1995.

- [321] D.R. Stinson. Some observations on the theory of cryptographic hash functions. See [EP], # 2001/020, 2002.
- [322] H.-M. Sun and B.-T. Hsieh. Security analysis of Shim's authenticated key agreement protocols from pairings. See [EP], # 2003/113, 2003.
- [323] E. Teske. Speeding up Pollard's rho method for computing discrete logarithms. In [A-3], 541–554.
- [324] E. Teske. An elliptic curve trapdoor system. *J. Cryptology*, To appear.
- [325] N. Thériault. Index calculus attack for hyperelliptic curves of small genus. In [A03], 75–92.
- [326] N. Thériault. Weil descent attack for Kummer extensions. *J. Ramanujan Math. Soc.*, **18**, 281–312, 2003.
- [327] N. Thériault. Weil descent attack for Artin-Schreier curves. Preprint, 2004.
- [328] É. Thomé. Subquadratic computation of vector generating polynomials and improvement of the block Wiedemann algorithm. *J. Symbolic Comput.*, **33**, 757–775, 2002.
- [329] E. Trichina and A. Bellezza. Implementation of elliptic curve cryptography with built-in countermeasures against side channel attacks. In [CH02], 98–113.
- [330] S. Vaudenay. Security flaws induced by CBC padding – Applications to SSL, IPSEC, WTLS... In [E02], 534–546.
- [331] S. Vaudenay. Hidden collisions on DSS. In [C96], 83–87.
- [332] J. Vélú. Isogénies entre courbes elliptiques. *C.R. Acad. Sc. Paris, Série A*, **273**, 238–241, 1971.
- [333] F. Vercauteren. *Computing Zeta Functions of Curves over Finite Fields*. PhD thesis, Katholieke Universiteit Leuven, 2003.
- [334] F. Vercauteren, B. Preneel and J. Vandewalle. A memory efficient version of Satoh's algorithm. In [E01], 1–13.
- [335] E.R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. In [E01], 195–210.
- [336] E.R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. *J. Cryptology*, To appear.
- [337] E.R. Verheul. Self-blindable credential certificates from the Weil pairing. In [A01], 533–551.
- [338] C.D. Walter. Montgomery's multiplication technique: How to make it smaller and faster. In [CH99], 80–93.
- [339] C.D. Walter. Sliding windows succumbs to Big Mac attack. In [CH01], 286–299.
- [340] C.D. Walter. Breaking the Liardet-Smart randomized exponentiation algorithm. In P. Honeyman, editor, *Smart Card Research and Advanced Applications*, 59–68. Usenix Association, 2002.
- [341] C.D. Walter. Some security aspects of the MIST randomized exponentiation algorithm. In [CH02], 276–290.
- [342] C.D. Walter and S. Thompson. Distinguishing exponent digits by observing modular subtractions. In D. Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, LNCS 2020, 192–207. Springer-Verlag, 2001.
- [343] L.C. Washington. *Elliptic Curves: Number Theory and Cryptography*. CRC Press, 2003.
- [344] E. Waterhouse. Abelian varieties over finite fields. *Ann. Sci. École Norm. Sup.*, 4th series, **2**, 521–560, 1969.
- [345] B.R. Waters, D. Balfanz, G. Durfee and D.K. Smetters. Building an encrypted and searchable audit log. In *Proc. of Network and Distributed System Security Symposium 2004 – NDSS '04*, 2004.

- [346] A. Weil. Numbers of solutions of equations in finite fields. *Bull. Amer. Math. Soc.*, **55**, 497–508, 1949.
- [347] A. Weil. The field of definition of a variety. *Am. J. Math.*, **78**, 509–524, 1956.
- [348] N. Weste and K. Eshraghian. *Principles of CMOS VLSI Design*. Addison-Wesley, 2nd ed., 1993.
- [349] P. Wright. *Spy Catcher: The Candid Autobiography of a Senior Intelligence Officer*. Viking Press, 1987.
- [350] X. Yi. An identity-based signature scheme from the Weil pairing. *IEEE Comm. Lett.*, **7**, 76–78, 2003.
- [351] X. Yi. Efficient ID-based key agreement from Weil pairing. *Elect. Lett.*, **39**, 206–208, 2003.
- [352] D.H. Yum and P.J. Lee. Efficient key updating signature schemes based on IBS. In K.G. Paterson, editor, *Cryptography and Coding*, LNCS 2898, 167–182. Springer-Verlag, 2003.
- [353] F. Zhang and X. Chen. Attack on two ID-based group key agreement schemes. See [EP], # 2003/259, 2003.
- [354] F. Zhang and K. Kim. ID-based blind signature and ring signature from pairings. In [A02], 533–547.
- [355] F. Zhang and K. Kim. Efficient ID-based blind signature and proxy signature from bilinear pairings. In R. Safavi-Naini, editor, *Information Security and Privacy – ACISP 2003*, LNCS 2727, 312–323. Springer-Verlag, 2003.
- [356] F. Zhang and S. Liu. ID-based one round authenticated tripartite key agreement protocol with pairings. See [EP], # 2002/122, 2002.
- [357] F. Zhang, R. Safavi-Naini and W. Susilo. Efficient verifiably encrypted signatures and partially blind signatures from bilinear pairings. In T. Johansson and S. Maitra, editors, *Progress in Cryptology – INDOCRYPT 2003*, LNCS 2904, 191–204. Springer-Verlag, 2003.
- [358] F. Zhang, R. Safavi-Naini and W. Susilo. An efficient signature scheme from bilinear pairings and its applications. In [P04], 277–290.
- [359] F. Zhang, R. Safavi-Naini and C.-Y. Lin. New proxy signature, proxy blind signature and proxy ring signature schemes from bilinear pairing. See [EP], # 2003/104, 2003.

Summary of Major LNCS Proceedings

For ease of reference we include here a table listing the main conference proceedings and the associated LNCS volume numbers. This includes all conferences in the relevant series which were published by Springer-Verlag and not necessarily those just referenced in this book.

Year	Crypto	Eurocrypt	Asiacrypt	CHES	PKC	ANTS
2004		3027			2947	3076
2003	2729	2656	2894	2779	2567	
2002	2442	2332	2501	2523	2274	2369
2001	2139	2045	2248	2162	1992	
2000	1880	1807	1976	1965		1838
1999	1666	1592	1716	1717	1560	
1998	1462	1403	1514		1431	1423
1997	1294	1233				
1996	1109	1070	1163			1122
1995	963	921				
1994	839	950	917			877
1993	773	765				
1992	740	658				
1991	576	547	739			
1990	537	473				
1989	435	434				
1988	403	330				
1987	293	304				
1986	263					
1985	218	219				
1984	196	209				
1982		149				

Author Index

- Abdalla, M., 12, 50
 Adleman, L., 142, 144
 Al-Riyami, S.S., 247, 248
 van Antwerpen, H., 230
 Appenzeller, G., 249
 Arbaugh, W.A., 249
 Atkin, A.O.L., 103

 Balasubramanian, R., 192, 208, 209
 Balfanz, D., 249
 Barreto, P.S.L.M., 205, 206, 211, 215
 Bellare, M., 12, 26, 41, 50, 240, 241
 Blake-Wilson, S., 241
 Bleichenbacher, D., 8, 26, 74
 Boldyreva, A., 233
 Boneh, D., 8, 42, 194, 195, 202, 215, 218,
 219, 221–233, 235, 239, 240, 244, 247,
 249, 250
 Boyd, C., 241
 Boyen, X., 234, 239
 Brezing, F., 211
 Brier, E., 88
 Brown, D., 31
 Bull, L., 233
 Burmester, M., 242

 Canetti, R., 32, 238, 239
 Cantor, D.G., 136
 Carls, R., 115
 Cha, J., 228
 Chaum, D., 97, 230
 Chen, A.H., 246
 Chen, L., 234, 241
 Chen, X., 247
 Cheon, J.H., 122, 123, 228
 Chevallier-Mames, B., 94
 Ciet, M., 94
 Cocks, C., 210, 221
 Coppersmith, D., 205, 231
 Coron, J.-S., 84, 85
 Couveignes, J.-M., 103

 Dahab, R., 93
 Dalton, C.R., 249
 DeMarrais, J., 142, 144
 Deneff, J., 132
 Dent, A., 32
 Desmedt, Y.G., 242
 Deuring, M., 105
 Diem, C., 231
 Dodis, Y., 240
 Dupont, R., 211, 219
 Durfee, G., 249
 Duursma, I., 208
 Dwork, B., 132

 Eisenträger, K., 206
 Elkies, N., 103
 Enge, A., 143, 144, 147, 211, 219

 Flassenberg, R., 144
 Fouquet, M., 113
 Franklin, M., 202, 215, 218, 219, 221–229,
 235, 240, 244, 247, 249, 250
 Frey, G., 151, 185, 189, 197
 Fujisaki, E., 225, 237, 247

 Galbraith, S.D., 205, 208, 242
 Garefalakis, T., 192
 von zur Gathen, J., 131
 Gaudry, P., 113, 115, 121, 132, 144, 147,
 148, 152, 156
 Gauss, F., 136
 Gentry, C., 233, 235, 237, 246, 247, 249
 Gerhard, J., 131
 Gerkman, R., 132
 Gligor, V.D., 246
 Goldreich, O., 32
 Goldwasser, S., 23, 26, 41
 Goubin, L., 84
 Gürel, N., 132

 Hahn, S.G., 122, 123
 Halevi, S., 32, 238, 239

- Harley, R., 103, 113, 115, 127, 128, 131, 148, 156
 Harrison, K., 205
 Herranz, J., 232
 Hess, F., 152, 189, 191
 Hopkins, H.J., 242
 Horowitz, J., 235
 Howgrave-Graham, N, 8, 26
 Huang, M.-D., 142, 144

 Izu, T., 206

 Johnson, D., 241
 Joux, A., 42, 202, 203, 215, 218, 220, 221, 223, 242
 Joye, M., 88, 90, 94

 Karatsuba, A., 103
 Kasahara, M., 215, 218, 220, 222, 228, 234, 237, 240, 241, 249
 Katz, J., 238–240, 249
 Kedlaya, K.S., 129, 132
 Kempf, J., 249
 Khalili, A., 249
 Kim, H.Y., 122, 123, 205
 Kim, J.H., 122, 123
 Kim, K., 232, 247
 Kobitz, N., 99, 104, 133, 192, 208, 209
 Kocher, P., 72, 73
 Kohel, D., 115, 201
 Kudla, C., 241

 Lagrange, J.-L., 136
 Lang, S., 210
 Lauder, A., 132
 Lauter, K., 206
 Law, L., 10
 Lee, H.-S., 208
 Lee, P.J., 240
 Lercier, R., 103, 126, 132
 Liardet, P.-Y., 90, 91
 Libert, B., 232–234
 Lichtenbaum, S., 185
 Lin, C.-Y., 232
 Liu, S., 242
 López, J., 93
 Lubicz, D., 126, 132
 Lubin, J., 105, 106
 Lynn, B., 194, 205, 206, 211, 228–235, 249

 Malone-Lee, J., 233, 234

 Mao, W., 241
 Maurer, U., 8
 Menezes, A., 10, 197, 198, 200, 241
 Messerges, T.S., 84
 Messing, W., 106
 Mestre, J.-F., 115, 132
 Micali, S., 23, 41
 Micciancio, D., 26
 Miller, V., 196, 201
 Mironov, I., 233
 Miyaji, A., 209, 240
 Moenck, R.T., 131
 Montgomery, P., 93, 206
 Morain, F., 211
 Müller, V., 144

 Nakabayashi, M., 209
 Nalla, D., 242
 Naor, M., 230, 237
 Nguyen, K., 193, 202
 Nguyen, P., 8, 26, 42

 Ofman, Y., 103
 Ohgishi, K., 215, 218, 220, 222, 228, 234, 237, 240, 241, 249
 Okamoto, T., 197, 198, 200, 225, 237, 247
 van Oorschot, P., 142

 Palacio, A., 233, 240
 Park, J.H., 122, 123
 Park, J.Y., 122, 123
 Paterson, K.G., 241, 247, 248
 Paulus, S., 144
 Pieprzyk, J., 233
 Pinch, R., 210
 Pointcheval, D., 22
 Pollard, J., 142, 148

 Qu, M., 10
 Quisquater, J.-J., 90, 232–234

 Rück, H.-G., 141, 185, 189, 197
 Rabin, M.O., 41
 Rackoff, C., 41
 Reddy, K.C., 242
 Rivest, R., 23
 Rogaway, P., 12, 41, 50, 241
 Rubin, K., 208

 Sáez, G., 232
 Safavi-Naini, R., 232, 233

- Sakai, R., 215, 218–220, 222, 228, 234,
 237, 240, 241, 249
 Satoh, T., 103–132
 Schönhage, A., 103
 Schoof, R., 103
 Scott, M., 205, 206, 211
 Serre, J.-P., 104–106, 132
 Shacham, H., 194, 229–233
 Shamir, A., 218, 221, 228
 Shanks, D., 137
 Shim, K., 241
 Shoup, V., 31, 57, 61, 62, 64, 233
 Shparlinski, I., 8, 26, 201, 242
 Silverberg, A., 208, 235, 237, 249
 Silverman, J.H., 184, 191, 198, 212
 Simon, D., 41
 Skjernaa, B., 113, 122–125, 129, 130
 Smart, N.P., 8, 26, 57, 90, 91, 152, 240,
 241
 Smetters, D.K., 249
 Soldera, D., 205
 Solinas, J., 10
 Stein, A., 143, 144
 Steinfeld, E., 233
 Stern, J., 22
 Stinson, D., 28
 Strassen, V., 103
 Susilo, W., 233

 Taguchi, Y., 122–125, 129, 130
 Takagi, T., 206
 Takano, S., 209
 Tate, J., 105, 106, 132, 185
 Thériault, N., 148, 149, 156
 Thiel, C., 144

 Vanstone, S., 10, 197, 198, 200
 Vaudenay, S., 27
 Vélú, J., 110
 Vercauteren, F., 114, 122, 132
 Verheul, E., 194, 203, 215, 216, 221, 226,
 227, 233

 Wan, D., 132
 Wang, H., 233
 Warinschi, B., 233
 Washington, L.C., 184
 Waterhouse, W.C., 199
 Waters, B.R., 249
 Weil, A., 136, 153, 185
 Weng, A., 211
 Wiener, M., 142

 Wolf, S., 8
 Wright, P., 72

 Yi, X., 241
 Yum, D.H., 240
 Yung, M., 240

 Zhang, F., 232, 233, 242, 247

Subject Index

- abelian variety, 151
- active attack, 64
 - on a device, 69, 71–72
- adaptive chosen ciphertext attack, *see* CCA2
- addition formulae
 - dummy operations, 91–92
 - indistinguishable, 88–92
 - unified, 88–89
- Advanced Encryption Standard, *see* AES
- advantage, 44
- AES, 12
- aggregate signature, 233
- AGM, 115–121
 - algorithm, 119–120
 - univariate, 120–121
- anomalous attack, 141
- ANSI, 18
 - ANSI X9.62, 4, 174
 - ANSI X9.63, 4
- Application protocol data units, 71
- Artin–Schreier
 - construction, 164
 - equation, 125–128, 154
 - extension, 153, 155, 176
 - operator, 164
- Baby Step/Giant Step, *see* BSGS
- BDH problem, 202, 203, 218, 219, 221, 222, 224, 226, 228, 234, 237, 238, 241, 242, 247, 250
 - generalized, 248
- benign malleability, 14, 15, 61
- bilinear Diffie–Hellman problem, *see* BDH problem
- bilinearity (of modified pairing), 217
- binary tree encryption, 238
- black box groups, 8
- blind signature, 97
- BLS short signature, 229–232
- Boneh–Franklin encryption scheme, 222–226
- BSGS, 18, 142
- canonical lift, 105–108, 116–117
- Cantor’s algorithm, 136, 140
- CBE, 246–248
- CCA, 16, 46, 48, 50, 51, 64–66, 74, 224, 225, *see also* CCA1 and CCA2
- CCA1, 46, 64
- CCA2, 13, 14, 46, 48, 61, 64
- CDH problem, 47, 48, 50, 202, 228–230, 232, 233, 237, 250
- Certicom, 4
- Certificate-Based Encryption, *see* CBE
- Certificateless Public Key Cryptography, *see* CL-PKC
- Certification Authority, 218
- chosen ciphertext attack, *see* CCA
- chosen plaintext attack, *see* CPA
- CL-PKC, 247–249
- CM method, 209, 210
- cofactor Diffie–Hellman, 9
- collusion resistant, 220
- complexity-theoretic, 47
- computational Diffie–Hellman problem, *see* CDH problem
- conorm, 152
- conversion function, 5, 24, 26, 29, 32, 33
- correlation analysis, 76
- CPA, 45, 46, 64
- cryptographic hardware, 69–71
- cryptographic workflow, 244–246, 248
- curve validation, 18
- cyclotomic polynomial, 210
- Data Encapsulation Mechanism, *see* DEM
- data origin authentication, 9
- DBDH problem, 202, 218, 234, 238, 239
- DDH problem, 47, 50, 55–58, 202, 229, 230
- decision bilinear Diffie–Hellman problem, *see* DBDH problem

- decision Diffie–Hellman problem, *see* DDH problem
- degree of a function, 212
- DEM, 15, 17, 62–66
- DHAES, 12
- DHIES, 12
- differential side-channel analysis
 - point multiplication, 84
- Diffie–Hellman problem, 14
- Diffie–Hellman protocol, 8–10, 221, 241, 242
- Digital Signature Algorithm, *see* DSA
- Digital Signature Scheme, *see* DSA
- distance-of-mean test, 76, 85
- distortion map, 194
- divisor, 184
 - class group, 152, 153, 184
 - defined over K , 184
 - degree, 134, 184
 - equivalent, 184
 - evaluation of function at, 185
 - group, 134
 - of a function, 184
 - of function, 134
 - principal, 134, 184
 - reduced, 135, 137, 139, 142–144, 146, 148, 149
 - weight, 135
 - smooth, 142–143, 149
 - support, 184
- domain parameters, 5, 6, 8, 12, 16
 - attack, 26, 27
- DSA, 4–7, 21, 229, 231
- dual isogeny, 166

- ECDDH problem, 202, *see also* DDH problem
- ECDH protocol, 4, 8–10, 18, 47, 48, 220, *see also* DH protocol
- ECDHP, 8, 202, 231, *see also* CDH problem
- ECDLP, 8, 151–179
- ECDSA, 4–9, 12, 21–40, 57, 77, *see also* DSA
- ECIES, 4, 12–18, 41–66
- ECIES-KEM, 4, 15–17, 61–66
- ECMQV, 4, 10–12, 18
- electromagnetic radiation leakage, 69, 74
- ElGamal encryption, 223
- elliptic curve cryptosystems
 - attacks on, 70
 - fault attacks, 72
 - side-channel analysis on, 70
- elliptic curves
 - constructing with given embedding degrees, 208–212
 - division polynomial, 109, 110, 112, 113
 - generating with CM method, 210
- embedding degree, 189
- endomorphism ring, 168
- ephemeral public keys, 8
- ephemeral secret, 7
- error-message attacks, 74
- exponent, 183
- external authenticate, 71

- fault attacks, 71–72
- FIPS
 - FIPS-140-1, 70
 - FIPS-186, 4
 - FIPS-186.2, 4
- forgery, 23
 - active, 24
 - existential, 24
 - passive, 24
 - selective, 24
- forgery, 23
- Forking Lemma, 22
- forward secrecy, 10
- forward secure encryption, 238
- Frey–Rück attack, 19, 141, 197–199
- Frobenius
 - automorphism, 152, 154, 160, 164, 165
 - endomorphism, 99–100, 136
 - map, 199
- FS-PKE, 238–240
- Fujisaki–Okamoto hybridization, 225, 237, 247
- FullIdent, 224
- function, 184
 - defined over K , 184
 - on a curve, 212
- function field, 134, 152

- Galois theory, 152
- gap Diffie–Hellman
 - group, 229, 230
 - problem, 47, 50, 54–56
- Gauss’s algorithm, 136, 137
- Gauss’s composition, 136
- Gaussian Normal Basis, 122, 126, 129

- generic group model, 7, 31–35, 56–58, 65, 141
- genus, 133, 135–137, 140–143, 146, 148–150, 153–157, 159, 160, 162, 165, 172–174
- GHS attack, 152–175
 - isogenies, 165–172
- GMR Security, 23–24
- GNB, 122
- GRH, 171
- Hagelin machine, 72
- Hamming weight, 205
- Harley’s algorithm, 127–128
- hash Diffie–Hellman problem, 50–54
- hash function
 - collision resistant, 28, 30
 - effective, 27, 32
 - one-way, 28, 30, 32
 - preimage-resistant, 28
 - second-preimage resistant, 28, 32
 - smooth, 30–31
 - uniform, 30–31
 - zero-resistant, 27, 32
- Hasse interval, 200
- Hasse’s Theorem, 103, 114
- HCDLP, 140–142, 151–179
 - index-calculus algorithm, 142, 144–150
- Hensel
 - lemma, 110
 - lifting, 110
- Hessian form, 90
- HIBE, 235–240
 - Gentry and Silverberg Scheme, 235–237
- hybrid encryption, 42, 61
- hyperelliptic curve, 133–150
 - group law, 136–140
 - Cantor’s algorithm, 136
 - Lagrange’s algorithm, 136
 - Jacobian, 134–135
- hyperelliptic involution, 134
- IBE, 215, 216, 221–230, 235, 237, 239, 240, 243–245, 247, 249, 250
- IBS, 228–230, 240
- ID based
 - blind signature, 232
 - encryption, 215, 218, 221–228, *see also* IBE
 - security of, 224–226
 - hierarchical cryptography, 235–240
 - key agreement, 240–242
 - non-interactive key distribution, 218–220
 - ring signature, 232
 - signatures, 218, 228–229, *see also* IBS
 - signcryption, 233–234
 - undeniable signature, 232
- ideal group model, *see* generic group model
- ideal hash model, *see* random oracle model
- IEEE 1363, 4
- IETF, 173
- IKE, 249
- ILA, 69
- IND-CCA2, 46, 50, 51, 54, 55, 57, 61, 64–66, 224, 226, 238, 239
- IND-ID-CCA, 225, 226
- index-calculus, 153, 156, 157, 159, 171, 173
- indistinguishability game, 43–46, 49, 50, 63, 64, *see also* IND-CCA2 and IND-ID-CCA
- information leakage analysis, 69
- internal authenticate, 71
- IPSec, 249
- ISO, 4
- isogeny, 165
 - class, 166–167
 - computing, 168
 - cycles, 106
 - dual, 166
- Jacobi form, 90–91
- Jacobian, 134–136, 142, 144, 148, 152
- Karatsuba multiplication, 103
- KEM, 15–17, 61–66
- KEM-DEM cipher, 17, 62–66
- key agreement
 - tripartite, 215
- key confirmation, 12
- key derivation function, 50, 51, 56
 - idealized, 54–56
- key distribution
 - Diffie–Hellman, 8–10
 - ECMQV, 10
 - EQMQV, 12
 - from pairings, 218–221, 240–242
 - multi-party, 242
 - non-interactive ID based, 218–220
 - tripartite, 220–221, 242
- Key Encapsulation Mechanism, *see* KEM

- key transport, 10
- Koblitz curve, 99, 149
- Kronecker relation, 107, 114
- Kronecker–Hurwitz class number, 166
- Kummer extension, 153, 175
- L-polynomial, 155
- Lagrange’s algorithm, 136, 138, 140, 146
- Lagrange’s Theorem, 98
- Lanczos’s algorithm, 146, 147, 150
- Lercier–Lubicz algorithm, 126–127
- lunchtime attacks, 46
- MAC, 12, 13, 15, 42, 48–54, 56, 57, 65, 218
- magic number, 155
- Markov chain, 80
 - aperiodic, 80
 - irreducible, 80
 - stationary distribution, 80
- Markov process, 80
 - hidden, 82
- meet-in-the-middle attack, 83
- MESD, 85
- Message Authentication Code, *see* MAC
- midnight attacks, 46
- Miller’s algorithm, 196–197, 205–207
- MNT criteria, 209–210
- Monsky–Washnitzer cohomology, 132
- MOV attack, 19, 141, 197–199, 231
- multiplicity, 184
- multiplier
 - blinding, 98–99
 - splitting, 99
- NIKDS, 218, 220, 222, 223, 228, 234
- NIST, 26
- non-degeneracy (of modified pairing), 217
- non-rational endomorphism, 194
- non-repudiation, 26, 39
- norm, 152
- normal basis, 158
- NUCOMP, 137–140, 146
- NUDPL, 139
- NUDUPL, 140
- one-way game, *see* OW game
- ordinary, 194, 198
- OW game, 43, 45, 46, 48
- pairing, *see also* Tate pairing and Weil pairing
 - bilinear, 183–184
 - bilinearity, 183
 - deducing group structure, 201
 - non-degeneracy, 183
 - properties of, 183, 216–218
 - protocols based on, 215–251
 - symmetry of, 195
- partial key exposure, 8, 26
- passive attack, 49, 64
 - on a device, 69, 72–77
- Pearson correlation coefficient, 76
- Pell equation, 209
- PKCS#1, 74
- Pohlig–Hellman simplification, 141
- point blinding, 97
- point counting, 103–132
- point multiplication
 - atomic, 94–97
 - binary, 79
 - double-and-add-always, 93
 - low Hamming weight, 206
 - Montgomery, 93–94
 - randomization techniques
 - base point, 97–98
 - multiplier, 98–100
 - window methods, 206
- Pollard methods, 152, 156, 157, 160, 170, 171, 173, 174
 - lambda method, 142
 - rho algorithm, 18, 148, 149
- power consumption leakage, 73
 - Hamming weight leakage, 73
 - transition count, 73
- private key generator, 219
- projective representation
 - randomized, 97–98
- provable security
 - signatures, 21–40
- public key validation, 18
- public-key encryption scheme
 - deterministic, 42
 - probabilistic, 42, 43
 - sound, 42
- Quadratic Residuosity problem, 221
- quaternion algebra, 198
- Rück attack, 141–142
- ramification index, 212
- ramification points, 134

- random oracle model, 32–36, 41, 54–57, 65, 219, 226, 228, 230
- random walks, 142
- randomized isomorphism
 - curve, 98
 - field, 98
- rarely zero hash, 27
- Riemann–Roch theorem, 135
- RSA, 9, 74, 97
- RSA-OAEP, 74

- Satoh’s algorithm, 103–132
- Satoh–Skjernaa–Taguchi algorithm, 123–125
- SCA, 69–100
- Schönhage–Strassen multiplication, 103
- Schoof’s algorithm, 103
- SEA algorithm, 103
- SECG, 4, 18
- security multiplier, 189
- self-pairings, 193
- SEMD, 84
- semi-logarithm, 24–26, 29, 35
- SHA-1, 5, 19
- SHA-256, 5
- SHA-384, 5
- side-channel analysis
 - simple, 87
- side-channel analysis, 8, 69–100
 - combining, 74
 - differential, 69, 75–76, 84
 - first-order, 76
 - multiple-exponent single-data, 85
 - point arithmetic, 80–83
 - point multiple, 77
 - second-order, 76
 - simple, 69, 74–75
 - point multiplication, 77–83
 - single-exponent multiple-data, 84
 - zero-exponent multiple-data, 85
- side-channels, 72–74
- smart cards, 71
 - simple attacks on, 71
- SSL/TLS protocol, 242
- straight line program, 197
- supersingular curve, 194, 198–201
 - embedding degrees, 199
- symmetric cipher, 48–218
- symmetric encryption, 50
- symmetry (of modified pairing), 217
- tamper attacks, 70, 71
- tamper resistant device, 70
- Tate pairing, 48, 141, 183, 185–197, 206–208, 216, 217, 250
 - efficient computation, 205–208
 - Miller’s algorithm, 196–197
 - over finite fields, 189–191
 - properties, 187–189
- timing attack, 72–73
- timing variation attacks, 72
- trace map, 194–195
- tripartite key agreement, 220–221, 223
- Trusted Authority, 218

- Vélu’s formulae, 110, 111, 113
- Vercauteren’s algorithm, 114–115
- Vernam cipher, 48, 49, 60
- Verschiebung, 109
- Viterbi algorithm, 83

- Weidemann’s algorithm, 150
- Weierstraß point, 133, 134
- Weil conjectures, 106, 119
- Weil descent, 151–179, 208, 231
 - Artin–Schreier constructions, 176–177
 - Kummer constructions, 175–176
- Weil pairing, 48, 141, 183, 185, 191–197, 201, 206, 207, 216, 217, 250
 - generalized, 192
 - properties, 191
- Weil reciprocity, 184–185, 212–213
- Weil restriction, 151
- Wiedemann’s algorithm, 146

- ZEMD, 85
- zeta function, 155