Advances in asymmetric cryptographic algorithms

Benjamin Smith

Synthèse d'Activité Scientifique

Advances in asymmetric cryptographic algorithms

Benjamin Smith

July 2023

Abstract

This document gives a synthetic overview of a large part of my research since 2006, for the degree of *Habilitation à Diriger des Recherches* at the Institut Polytechnique de Paris.

We focus on algorithms for asymmetric (or public-key) cryptography—especially, though not exclusively, *Elliptic Curve Cryptography*; its generalizations including *Hyperelliptic Curve Cryptography*; and its derivatives including *Isogeny-Based Cryptography*, one of the newer branches of post-quantum cryptography. We also consider the implementation of post-quantum cryptosystems in a broader sense. The contributions range from theoretical and mathematical foundations through to low-level implementation techniques.

Habilitation à Diriger des Recherches
Defence: 6th of October, Campus Cyber, La Défense Paris, France

Emmanuel Thomé, Inria Nancy & LORIA, Président
Pierre-Alain Fouque, Université Rennes 1, Rapporteur
Florian Hess, Carl on Ossietzky Universität Oldenburg, Rapporteur
Alfred Menezes, University of Waterloo, Rapporteur
Catuscia Palamidessi, Inria Saclay & LIX, Examinatrice
Nigel Smart, Katholiek Universiteit Leuven, Examinateur

Contents

1 Introduction and highlights

This document gives a synthetic overview of my research since 2006. It is focused on cryptographic algorithms: especially, though not exclusively, *Elliptic Curve Cryptography* (ECC), its generalizations, and its derivatives.

1.1 Context

My research exploits algebraic geometry and number theory to attack and improve cryptosystems, especially those built around elliptic curves and abelian varieties (higher-dimensional generalizations of elliptic curves, notably Jacobians of *hyperelliptic curves*). The fundamental tools are explicit *isogenies*, which are algebraic transformations of abelian varieties, and *endomorphisms*, which are transformations of an abelian variety into itself.

ECC, introduced by Miller and Koblitz in 1985 [Mil85; Kob87] is now crucial to internet security: it is the state-of-the-art for *key exchange* (establishing shared cryptographic keys), and offers the most compact *digital signatures*. For industrial applications we target a minimum *security level* of 128 bits: adversaries require roughly 2^{128} operations (far beyond the limits of feasibility) to have any non-negligible probability of breaking the scheme. At the 128-bit security level, ECC public keys are only 256 bits long—twelve times shorter than the equivalent 3072-bit RSA keys. Working with smaller keys always reduces bandwidth, storage, and infrastructure costs for key distribution. Often, it even saves time and energy: ECC is substantially faster than RSA for key establishment and public-key encryption.

Conventional ECC is based on the supposed hardness of the elliptic-curve DLP. But powerful general-purpose quantum computers would render ECC (and indeed, virtually all public-key cryptosystems currently deployed on the internet) completely insecure: Shor's famous quantum algorithm [Sho97] solves DLP instances in polynomial time. Cryptographic research is therefore turning towards *post-quantum* cryptosystems, which run on conventional computers yet resist quantum-equipped adversaries. ECC is finding a post-quantum future in *isogeny-based cryptography* (IBC). ECC deals with points on a single curve $\mathscr{E}$: given P and $Q = [m]P$ on $\mathscr{E}$, the hard problem is to compute the algebraic relationship between them, which is the discrete logarithm m. In IBC, points are irrelevant: instead, we deal with multiple curves in a single class, and the hard problem is to compute the algebraic relationship between curves $\mathscr{E}$ and $\mathscr{E}'$: that is, an isogeny.

In real-world applications, reducing the computational cost of cryptography is critical. Faster cryptography means reduced latency and server costs, and improved battery lifetime for mobile computing. Reducing memory footprints becomes crucial when we move to low-end IoT devices operated by low-cost microcontrollers where very little RAM is available (even less once the enveloping ap-

plication is taken into account). Like ECC, IBC generally promises small keys and low bandwidth, but at the cost of intensive algebraic calculations. Compactness makes ECC and IBC interesting in constrained devices, where reducing the intensive computations, and providing even the most basic protections against side-channel attacks, is a fascinating scientific challenge. My work here has spilled over into compact implementations of non-curve-based public-key systems.

1.2 Organisation, highlights, and collaborations

Chapter 2 describes some of my results in **classical ECC**, setting the scene for later chapters. In [Smi16] and [Smi13], I introduced a new family of elliptic curves with efficient endomorphisms, which can be used to accelerate scalar multiplication following the Gallant–Lambert–Vanstone method (§2.4). François Morain, Charlotte Scribot,[1] and I used these endomorphisms to accelerate point-counting algorithms [MSS16] (§2.6). Craig Costello, Huseyin Hisil, and I used one family from [Smi16; Smi13] to develop free, high-speed, and side-channel protected software for Diffie–Hellman key exchange [CHS14] (§2.5).

Chapter 3 surveys my work in **hyperelliptic-curve cryptography**. My doctoral thesis with David Kohel [Smi06] developed algorithms for hyperelliptic isogenies and endomorphisms (§3.2), and exploited these to speed up scalar multiplication [KS06] (§3.3). David Kohel, Pierrick Gaudry, and I developed faster point-counting algorithms for families of curves with efficient endomorphisms in [GKS11], computing cryptographic parameters (and finding secure curves) far more rapidly in practice (§3.4). Over a decade later, [GKS11] still holds the world record in genus-2 point counting; this paper won the Best Paper award at Asiacrypt 2011. Hyperelliptic point counting *without* efficient endomorphisms remains extremely slow. Samuel Dobson, Steven Galbraith, and I have proposed higher-dimensional Jacobians *without* efficient endomorphisms as a source of cryptographic unknown-order groups with trustless setup [DGS22] (§3.6).

Isogenies also have destructive applications: my genus-3 attack [Smi08] was a totally new approach to DLP transfer, translating a technical and abstract result from algebraic geometry (Recillas' trigonal construction) into explicit formulas and algorithms for cryptographers (§3.5). This result essentially killed genus 3 in DLP-based cryptography; [Smi08] won the best paper award at Eurocrypt 2008, and [Smi09] was an invited submission to Journal of Cryptology.

Chapter 4 covers ***commutative* isogeny-based cryptography**, which is based on actions of ideal class groups on isogeny classes of elliptic curves.[2] My contributions to the foundations of group-action-based IBC (§4.2) include the survey article [Smi18], an invited contribution to WAIFI 2018, and a proof of the quantum equivalence of the group-action analogues of the Discrete Logarithm and Computational Diffie–Hellman Problems [Gal+21] with Steven Galbraith, Lorenz Panny, and Frederik Vercauteren. Luca De Feo, Jean Kieffer, and I tested the practical limits of group-action cryptography with *ordinary* curves in [DKS18],[3] exposing serious parameterization problems at practical security levels. This led Castryck, Lange, Martindale, Panny, and Renes to switch from ordinary to *supersingular* curves with CSIDH [Cas+18]. Mathilde Chenu and I showed that the families of

[1] [MSS16] was the result of Charlotte Scribot's masters internship with us.
[2] Commutative IBC appears immune to the devastating recent attacks on SIDH and SIKE.
[3] [DKS18] was the subject of Jean Kieffer's masters internship with us.

curves from [Smi16] admitted class group actions in [CS22], extending and generalizing CSIDH and its derivatives (§4.2).[4] Gustavo Banegas, Valerie Gilchrist, and I explored fast supersingularity testing for CSIDH key-validation [BGS22][5] (§4.3).

Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and I created the *Vélusqrt* algorithm [Ber+20][6] to compute isogenies in square-root instead of linear time (§4.4). This is the first improvement in complexity for prime isogeny computation since the 1970s; it immediately improves any IBC scheme using large-prime-degree isogenies, including CSIDH. The CTIDH project [Ban+21a] set new speed records for CSIDH through Vélusqrt and a fundamental reconception of the private keyspace, while improving resistance to side-channel attacks (§4.5).

Moving to higher dimensions, Wouter Castryck, Thomas Decru, and I designed a genus-2 isogeny-based hash function [CDS20], before Craig Costello and I substantially reduced its security with a new high-dimensional isogeny-finding algorithm [CS20] (with serious implications for the (in)security of higher-dimensional abelian varieties in IBC). Enric Florit and I investigated the structure and random-walk properties of the genus-2 superspecial isogeny graph [FS21a; FS21b][7] (§4.6).

Chapter 5 concerns implementations of **public-key cryptosystems for constrained environments**. We begin with an example from hyperelliptic cryptography: Joost Renes, Peter Schwabe, Lejla Batina, and I used Kummer surfaces (the genus-2 analogue of the x-coordinate on elliptic curves) to produce compact, high-speed key exchange and signature software for microcontroller platforms with very low memory requirements [Ren+16; RS17] (§5.1).

The Inria *Défi* RIOT-FP [RFP] aims to provide "future-proof" security for RIOT [Bac+18; RIO], a free and open-source operating system for low-end IoT devices. In this context, Gustavo Banegas, Adrian Herrmann, Koen Zandberg, Emmanuel Bacceli, and I evaluated postquantum signature schemes in the context of secure software updates with the SUIT protocol in [Ban+22] (§5.2). This work was selected for presentation at Real-World Cryptography 2022.

Most recently, Gustavo Banegas, Thomas Debris-Alazard, Milena Nedeljković, and I developed the first implementation of WAVE, a post-quantum code-based signature scheme with fast verification on microcontrollers [Ban+21b].[8] I have continued this work in the WAVE submission [Ban+23a] to the NIST signature on-ramp for post-quantum cryptography standardization [NIST-sigs].

1.3 Other topics

Some of my cryptographic research is not covered in this document: notably, pairings in genus 2 [Gal+09], Weil descent attacks on binary elliptic curves with special endomorphisms [CRS21], eliminating semi-abelian varieties as candidate cryptographic groups [GS06], classical improvements to quantum factoring algorithms [Gro+15], and related algorithms for deterministic factoring with number-theoretic oracles [MRS21]. We also set aside some results from explicit number theory including low-degree isogenies in genus 2 [Smi12], sporadic isogeny families in higher genus [Smi10; Smi11], and arithmetic statistics in genus 3 [Ler+19].

[4] [CS22] includes the main results of Mathilde Chenu's PhD thesis under my direction.
[5] [BGS22] was a result of Valerie Gilchrist's pre-doctoral internship with us.
[6] [Ber+20] formed part of Antonin Leroux's prize-winning PhD with Luca De Feo and me.
[7] [FS21a] and [FS21b] were results from Enric Florit's outstanding pre-doctoral internship.
[8] [Ban+21b] includes the results of Milena Nedeljković's first-year summer internship with me on WAVE signature compression.

2 Classical elliptic-curve cryptography

We begin with a quick review of elliptic curves for cryptography in §§2.1-2.3. Then we summarize some contributions: §2.4 describes an original family of curves equipped with efficient endomorphisms, §2.5 describes high-speed key-exchange software based on one of these curves, and §2.6 describes a special point-counting algorithm for curves in the family. (For further background, see [Sil09; Cas91; Gal12; CS18].)

2.1 Elliptic curves for cryptography

An elliptic curve is, by definition, a curve of genus 1 with a distinguished rational point. More concretely, elliptic curves are realised by nonsingular plane cubics: for example, in characteristic $p \neq 2$ we have the *short Weierstrass model*

$$\mathscr{E} : y^2 = x^3 + ax + b \tag{2.1}$$

where $4a^3 + 27b^2 \neq 0$, or the *Montgomery model*

$$\mathscr{E} : By^2 = x(x^2 + Ax + 1) \tag{2.2}$$

where $A^2 \neq 4$ and $B \neq 0$. (The inequalities ensure that the curve is nonsingular.)

For cryptography, the coefficients a and b (or A and B) are in a finite field $\mathbb{F}_q$, where $q = p^d$ is a prime power; typically p is large and $d = 1$ or 2, though the "binary" case where $p = 2$ and d is prime has also been of interest in the past, especially where dedicated hardware for finite field arithmetic is involved.

Points on $\mathscr{E}$ are solutions $(x, y) = (\alpha, \beta)$ to the equation of $\mathscr{E}$, with α and β in $\mathbb{F}_q$, together with the unique "point at infinity" $\mathscr{O}$. The points form a commutative group $\mathscr{E}(\mathbb{F}_q) \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ for some m and n with $m \mid \gcd(n, q-1)$ and $|\#\mathscr{E}(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$. The group law is geometric, and therefore algebraic: it can be efficiently computed by a series of polynomial operations over $\mathbb{F}_q$.

To make the group law explicit for the Montgomery model in (2.2) above: $\mathscr{O}$ is the zero element; negation is $\ominus : (x : y : 1) \mapsto (x : -y : 1)$; and for addition, if $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ are points on $\mathscr{E}$, then $P \oplus Q = (x_\oplus, y_\oplus)$ where

$$\begin{cases} x_\oplus = B\lambda^2 - (x_P + x_Q) - A \\ y_\oplus = \lambda(x_P - x_\oplus) - y_P \end{cases} \quad \text{where} \quad \lambda = \begin{cases} \frac{y_Q - y_P}{x_Q - x_P} & \text{if } P \neq Q \text{ or } \ominus Q\,, \\ \frac{(3x_P^2 + 2Ax_P + 1)}{2By_P} & \text{if } P = Q\,; \end{cases}$$

if $P = \ominus Q$, then $P \oplus Q = \mathscr{O}$. We observe that λ is the slope of the secant through P and Q (or the tangent to $\mathscr{E}$ at P, in the case $P = Q$).

In cryptographic applications, the critical operation is *scalar multiplication*:

$$\text{Given } P \in \mathscr{E}(\mathbb{F}_p) \text{ and } m \in \mathbb{Z}\text{, compute } [m]P := P + \cdots + P \quad (m \text{ copies of } P).$$

For each positive integer m, the *m-torsion subgroup* is $\mathscr{E}[m] := \{P \in \mathscr{E} : [m]P = \mathscr{O}\}$. We say $\mathscr{E}$ is *supersingular* if $\mathscr{E}[p](\overline{\mathbb{F}}_q) = 0$; otherwise, $\mathscr{E}$ is *ordinary*.

In theory, scalar multiplication is efficient: computing $[m]P$ takes $O(\log m)$ group operations in $\mathscr{E}(\mathbb{F}_p)$. Optimizing scalar multiplication in practice has long been a subject of intense and competitive research. The simplest scalar multiplication algorithm is *double-and-add*. Start with $R = P$; then, for each bit in the binary expansion of m (from the most significant bit down), double R, and add P if the bit is 1. Generalizing from the binary expansion to 2^w-adic expansions yields a family of *fixed-window* scalar multiplications.

The *Discrete Logarithm Problem* (DLP) is the inverse of scalar multiplication:

$$\text{Given } P \text{ and } Q \in \mathscr{E}(\mathbb{F}_p) \text{ with } Q = [m]P \text{ for some } m\text{, compute } m.$$

For general $\mathscr{E}(\mathbb{F}_p)$, the fastest DLP algorithms known run in time $O(\sqrt{N})$, where N is the largest prime divisor of $\#\mathscr{E}(\mathbb{F}_p)$; so for prime-order curves, these algorithms solve DLPs in $O(\sqrt{p})$ by Hasse's theorem, and the DLP is exponentially hard (in the input size, which is in $O(\log p)$).

Once we get beyond the algebraic geometry, elliptic curve cryptosystems are generally quite simple. The simplest is Diffie–Hellman key exchange, which will serve to illustrate all the relevant ideas.

Alice and Bob want to establish a common encryption key over a public channel, without prior contact. We have a public, fixed elliptic curve $\mathscr{E}$ and a point P in $\mathscr{E}(\mathbb{F}_p)$. Alice samples a secret integer a, computes $A = [a]P$, and publishes A; Bob samples a secret integer b, computes $B = [b]P$, and publishes B. Then, Alice computes $S = [a]B$ and Bob computes $S = [b]A$; they have computed the same S because $[a][b]P = [b][a]P = [ab]P$.

The public keys $A = [a]P$ and $B = [b]P$ represent instances of the DLP in $\mathscr{E}(\mathbb{F}_p)$: to recover a private key, we must solve a DLP instance, so $\mathscr{E}$ and p must be chosen such that this is computationally infeasible. Recovering S from (P, A, B) is the *Computational Diffie–Hellman Problem* (CDHP). The CDHP is not obviously equivalent to the DLP, but there exist polynomial-time reductions under strong number-theoretic hypotheses, and subexponential-time reductions under more plausible ones [Mau94; MW99; MSV04; Ben05]. In practice, all known CDHP algorithms are DLP algorithms. So: if the DLP is hard in $\mathscr{E}(\mathbb{F}_p)$ then S is regarded as a shared secret, and Alice and Bob may derive a common secret encryption key from it.

The hard work for Alice and Bob is all in scalar multiplication. The scalars are secret keys, so the scalar multiplication must be done in (cryptographic) *constant time*: this means that the execution path, including the sequence of instructions and memory access locations, must be independent of the scalars so as to reduce leakage of secret values to side channels. For this we use the *Montgomery ladder* [Mon87; CS18], which computes scalar multiplication "up to sign": that is,

$$(x(P), m) \longmapsto x([m]P).$$

The ladder does not use y-coordinates, or the full elliptic-curve group law: instead, we use (projective versions of) the relations

$$x_{P\oplus Q}x_{P\ominus Q}(x_P - x_Q)^2 = (x_P x_Q - 1)^2 \quad \text{if } P \neq Q\,, \tag{2.3}$$

$$4x_{[2]P}x_P(x_P^2 + Ax_P + 1) = (x_P^2 - 1)^2\,. \tag{2.4}$$

If $y([m]P)$ is required, then it is easily recovered from $y(P)$, $x([m]P)$, and $x([m+1]P)$, which is in fact an auxiliary output of the ladder.

2.2 Isogenies and endomorphisms

We do not speak of vector spaces without speaking of linear transformations and matrices, or of groups without speaking of homomorphisms. Similarly, we cannot go far with elliptic curves without introducing the morphisms beween them.

A *homomorphism* of elliptic curves over $\mathbb{F}_q$ is an algebraic map $\phi : \mathscr{E}_1 \to \mathscr{E}_2$ (defined by coherent systems of tuples of polynomials, or rational functions, with coefficients in $\mathbb{F}_q$) mapping $\mathscr{O}_{\mathscr{E}_1}$ to $\mathscr{O}_{\mathscr{E}_2}$. Such a mapping is necessarily a homomorphism in the usual sense: it respects the group laws on $\mathscr{E}_1$ and $\mathscr{E}_2$, and induces a homomorphism $\mathscr{E}_1(\mathbb{F}_q) \to \mathscr{E}_2(\mathbb{F}_q)$ (see for example [Sil09, Theorem III.4.8]).

An *isogeny* is a homomorphism[1] $\phi : \mathscr{E}_1 \to \mathscr{E}_2$ with finite kernel that is (geometrically) surjective. For example, if $\mathscr{E} : y^2 = x^3 + a_2x^2 + a_4x + a_6$ is a curve over $\mathbb{F}_q$, then there is a p-power *Frobenius isogeny*

$$\pi_p : (x, y) \longrightarrow (x^p, y^p)$$

from $\mathscr{E}$ to the Galois-conjugate curve

$${}^{(p)}\mathscr{E} : y^2 = x^3 + a_2^p x^2 + a_4^p x + a_6^p\,.$$

An isogeny is *separable* if it does not factor through π_p. If S is a finite subgroup of $\mathscr{E}$, then there exists a curve $\mathscr{E}/S$ and a separable isogeny $\phi : \mathscr{E} \to \mathscr{E}/S$ with kernel S; *Vélu's formulæ* [Vél71] compute $\mathscr{E}/S$ and ϕ (see [Koh96, §2.4]).

An *endomorphism* is a homomorphism from a curve to itself. The simplest examples are the scalar multiplications $[m]$ and the *Frobenius endomorphism*

$$\pi_q : (x, y) \longmapsto (x^q, y^q)\,.$$

The endomorphisms of $\mathscr{E}$ form a ring, $\mathrm{End}(\mathscr{E})$: composing with $\oplus$ on $\mathscr{E}$ induces addition on endomorphisms, and multiplication is composition of endomorphisms. The structure of $\mathrm{End}(\mathscr{E})$ is a fundamental invariant of $\mathscr{E}$; it is used, implicitly or explicitly, in many of the algorithms discussed here. Any isogeny $\phi : \mathscr{E}_1 \to \mathscr{E}_2$ induces a close relationship between $\mathrm{End}(\mathscr{E}_1)$ and $\mathrm{End}(\mathscr{E}_2)$: in particular, the endomorphism algebras $\mathrm{End}(\mathscr{E}_1) \otimes \mathbb{Q}$ and $\mathrm{End}(\mathscr{E}_2) \otimes \mathbb{Q}$ are isomorphic.

2.3 Endomorphisms and scalar decomposition

Since elliptic curve scalar multiplication is analogous to exponentiation in finite fields, many algorithms originally developed for multiplicative groups transfer directly to scalar multiplication: finite-field square-and-multiply becomes elliptic-curve double-and-add, for example. But the geometry of elliptic curves can offer

[1] Equivalently, we can define elliptic isogenies to be nonzero homorphisms; but when we move to higher dimensions, there are nonzero homomorphisms that are not isogenies.

us new algorithms with no true finite field analogues. A spectacular (and easy) example of this phenomemon is scalar multiplication with endomorphism decompositions, originally proposed by Gallant, Lambert, and Vanstone [GLV01].

Let $\mathscr{E}/\mathbb{F}_q$ be an elliptic curve with a cyclic subgroup $\mathscr{G} \subset \mathscr{E}(\mathbb{F}_q)$ of order N. Given an integer $0 \leq m < N$ our goal is to evaluate $[m]$ on $\mathscr{G}$ as quickly as possible.

Let ψ be an endomorphism of $\mathscr{E}$ such that $\psi(\mathscr{G}) \subseteq \mathscr{G}$ (this is typical in cryptographic applications, where N is so close to $\#\mathscr{E}(\mathbb{F}_q)$ that there is no room for the image of $\mathscr{G}$ to be anything but $\mathscr{G}$ itself.) Then ψ restricts to an endomorphism of $\mathscr{G}$; since $\mathscr{G}$ is cyclic, its endomorphism ring is isomorphic to $\mathbb{Z}/N\mathbb{Z}$; and so ψ acts on $\mathscr{G}$ as multiplication by an integer *eigenvalue* $-N/2 < \lambda \leq N/2$ such that

$$\psi|_{\mathscr{G}} = [\lambda]|_{\mathscr{G}}.$$

A *decomposition* of m is any pair (a, b) such that

$$m \equiv a + b\lambda \pmod{N}.$$

Given a decomposition (a, b) of m, we can compute $[m]P$ for any P in $\mathscr{G}$ using a *multiexponentiation* algorithm[2] on $(P, \psi(P))$ to compute $[m]P$ as

$$[m]P = [a]P \oplus [b]\psi(P).$$

For example: given $(P, \psi(P))$, Straus' algorithm [Str64] computes $[a]P \oplus [b]\psi(P)$ with $\log_2 \max(|a|, |b|)$ doubles and (on average) $\frac{3}{4}\log_2 \max(|a|, |b|)$ adds, while traditional 1-dimensional double-and-add computes $[m]P$ with $\log_2 |m|$ doubles and (on average) $\frac{1}{2}\log_2 |m|$ adds.

For scalar decomposition to offer an advantage over basic scalar multiplication, the endomorphism ψ and its eigenvalue λ must satisfy two conditions:

C1 *ψ must be efficiently computable*—that is, $P \mapsto \psi(P)$ should cost no more than a few group operations—to ensure the cost of initialising the multiexponentiation does not overwhelm its benefits.

C2 *λ must be large*, to ensure $\max(|a|, |b|) \ll |m| \approx N$ so multiexponentiation is faster than basic scalar multiplication. If $|\lambda| > \sqrt{N}$, then we can achieve $\max(|a|, |b|) \in O(\sqrt{N})$ (and this is optimal); finding such (a, b) is easy, and almost trivial for the ψ and λ in our applications [Smi15].

Achieving **C1** and **C2** simultaneously is hard: for general $\mathscr{E}/\mathbb{F}_q$, the only endomorphisms we know are linear combinations of $[1]$ and π_q. While $[1]$ and π_q are obviously efficient, both have eigenvalue 1. Any combination meeting **C1** will have tiny coefficients, and thus a tiny eigenvalue, violating **C2**; and conversely, any combination meeting **C2** violates **C1**.

We therefore need a supply of elliptic curves over $\mathbb{F}_q$ equipped with efficiently-computable endomorphisms that are *not* small linear combinations of $[1]$ and π_q. Gallant, Lambert, and Vanstone proposed using what are essentially reductions modulo p of CM curves over number fields.

[2] The literature on this topic is vast, and we will not attempt to summarize it here; for an introduction, we recommend [Knu97, §4.6.3], [Gal12, §2.8, §11.2] and [Ava+06, Chapter 9]).

Example 1 (GLV)**.** Suppose $p \equiv 1 \pmod 4$, so -1 is a square in $\mathbb{F}_p$. The curve

$$\mathscr{E} : y^2 = x^3 + x$$

has a highly efficient endomorphism

$$\psi : (x, y) \longmapsto (-x, \sqrt{-1}y)$$

and $\psi^2 = [-1]$. Clearly ψ meets **C1**. It also meets **C2**: given a subgroup $\mathscr{G} \subset \mathscr{E}(\mathbb{F}_p)$ as above, the eigenvalue λ is a square root of -1 modulo N, so $|\lambda| \geq \sqrt{N-1}$.

More generally, we can use any $\mathscr{E}/\mathbb{F}_q$ equipped with a *very* low-degree non-integer endomorphism (see [GLV01] for examples). The problem is that there are very few isomorphism classes of curves with such endomorphisms. If p is fixed—for example, if p is a special prime facilitating high-speed field arithmetic—then perhaps none of these curves have secure subgroups.

Galbraith, Lin, and Scott showed in [GLS11] that we can overcome the scarcity of secure GLV curves over $\mathbb{F}_p$ for fixed p by using special curves over $\mathbb{F}_{p^2}$.

Example 2 (GLS)**.** Fix a nonsquare δ in $\mathbb{F}_{p^2}$ and let $\mathscr{E}/\mathbb{F}_{p^2}$ be any curve in the form

$$\mathscr{E}/\mathbb{F}_{p^2} : y^2 = x^3 + (\delta^2 a_0)x + (\delta^3 b_0) \quad \text{with} \quad a_0, b_0 \in \mathbb{F}_p\,.$$

Now $\tau : (x, y) \mapsto (u, v) = (\delta^{-1}x, \delta^{-3/2}y)$ is an isomorphism of $\mathscr{E}$ onto $\mathscr{E}_0/\mathbb{F}_{p^2} : v^2 = u^3 + a_0 u + b_0$, which is the base-extension to $\mathbb{F}_{p^2}$ of a curve over $\mathbb{F}_p$, and as such has a p-power Frobenius endomorphism $\pi_p : \mathscr{E}_0 \to \mathscr{E}_0$. The twisted endomorphism

$$\psi := \tau^{-1} \circ \pi_p \circ \tau : (x, y) \longmapsto (\delta^{1-p}x^p, \delta^{3(1-p)/2}y^p) \quad \text{in } \mathrm{End}(\mathscr{E})$$

is defined over $\mathbb{F}_{p^2}$ because $2 \mid (p-1)$, and meets **C1** (especially with a good choice of δ: if $p \equiv 5 \pmod 8$, then $\delta = \sqrt{2}$ yields $\psi : (x, y) \mapsto (-x^p, \sqrt{-1}y^p)$)). It also meets **C2**: $\pi_{p^2}\tau = -\tau\pi_{p^2}$, so

$$\psi^2 = -\pi_{p^2}\,,$$

so λ is a square root of -1 on any $\mathscr{G} \subset \mathscr{E}(\mathbb{F}_{p^2})$ fixed by ψ, hence $|\lambda| \geq \sqrt{N-1}$.

This construction gives us $O(p)$ isomorphism classes of curves over any $\mathbb{F}_{p^2}$, so we can find secure GLS curves for any given p. However, we cannot find GLS curves $\mathscr{E}/\mathbb{F}_{p^2}$ such that $\mathscr{E}$ and its quadratic twist $\mathscr{E}_0$ *simultaneously* have secure subgroups, because $\mathscr{E}_0$ is a subfield curve. Twist-security is an important property for efficient elliptic-curve Diffie–Hellman constructions (see [Ber06a], [Fou+08]).

2.4 The $\mathbb{Q}$-curve construction

The $\mathbb{Q}$-curve construction of [Smi13] and [Smi16] is a generalization of GLV and GLS allowing a large choice of twist-secure curves when p is fixed.

Fix a nonsquare Δ in $\mathbb{F}_p$, so $\mathbb{F}_{p^2} = \mathbb{F}_p(\sqrt{\Delta})$. As we noted above, every

$$\mathscr{E}/\mathbb{F}_{p^2} : y^2 = x^3 + a_2x^2 + a_4x + a_6$$

has Frobenius isogenies $\pi_p : (x, y) \mapsto (x^p, y^p)$ to and from its Galois conjugate

$${}^{(p)}\mathscr{E} : y^2 = x^3 + a_2^p x^2 + a_4^p x + a_6\,.$$

Suppose there exists an isogeny $\phi : \mathscr{E} \to {}^{(p)}\mathscr{E}$ of very small degree d. Composing with $\pi_p : {}^{(p)}\mathscr{E} \to \mathscr{E}$ yields an endomorphism $\psi : \mathscr{E} \to \mathscr{E}$ of degree dp. Now ψ meets **C1** because $d = \deg(\phi)$ is tiny and π_p can be evaluated almost for free on points of $\mathscr{E}(\mathbb{F}_{p^2})$ because p-th powering amounts to changing the sign on $\sqrt{\Delta}$.

Curves over $\mathbb{F}_{p^2}$ with d-isogenies to their conjugates are parametrized by the $\mathbb{F}_p$-points of an Atkin–Lehner quotient of the modular curve $X_0(d)$; in particular, for $d = 2, 3, 5$, and 7 this quotient is rational, and we get one-parameter families of curves with efficiently computable endomorphisms. (The case $d = 1$, where ϕ is an isomorphism, is isomorphic to GLS.) In [Smi13] and [Smi16], this is expressed in terms of reductions of *quadratic* $\mathbb{Q}$*-curves*—that is, elliptic curves defined over quadratic number fields with isogenies to their conjugates—modulo inert p. The curves of Example 3 were constructed in this way from Hasegawa's universal family of degree-2 quadratic $\mathbb{Q}$-curves [Has97, Theorem 2.2].

Example 3 ([Smi16, §5])**.** Fix a nonsquare Δ in $\mathbb{F}_p$ and consider the family

$$\mathscr{E}_{2,s}/\mathbb{F}_{p^2} : y^2 = x^3 + 2(C_2(s) - 24)x - 8(C_2(s) - 16) \quad \text{where} \quad C_2(s) := 9(1 + s\sqrt{\Delta})$$

for s in $\mathbb{F}_p$. There is a 2-isogeny $\phi_{2,s} : \mathscr{E}_{2,s} \to {}^{(p)}\mathscr{E}_{2,s}$ with kernel $\langle (4,0) \rangle$, defined by

$$\phi_{2,s} : (x, y) \longmapsto \left(\frac{-x}{2} - \frac{C_2(s)}{x-4}, \frac{y}{\sqrt{-2}} \left(\frac{-1}{2} + \frac{C_2(s)}{(x-4)^2} \right) \right).$$

The composition $\psi_{2,s} := \pi_p \circ \phi_{2,s}$ satisfies **C1**—and also **C2**, because

$$\psi_{2,s}^2 = [\epsilon_p 2]\pi_{\mathscr{E}_{2,s}}, \quad \text{where} \quad \epsilon_p := -(-2/p) = \begin{cases} -1 & \text{if } p \equiv 1, 3 \pmod 8, \\ 1 & \text{if } p \equiv 5, 7 \pmod 8. \end{cases}$$

2.5 High-speed key exchange software

In [CHS14], we present high-speed Diffie–Hellman key exchange software based on the Montgomery model of a curve from the family in Example 3 over $\mathbb{F}_{p^2} = \mathbb{F}_p(\sqrt{-1})$ where p is the Mersenne prime $2^{127} - 1$ (this allows *very* fast field arithmetic). The curve parameters and their selection are detailed in [CHS14, §2].

The chief novelty here is the combination of the new endomorphism with x-only Montgomery arithmetic. This means multiexponentiation using a 2D *differential* addition chain: we tried 2D chains from Montgomery [Mon87], Azarderakhsh and Karabina [AK14], and Bernstein [Ber06b]. To initialise these chains, we need to efficiently compute the x-line action not only of ψ, but also of $\psi - 1$ (see [CHS14, §3]). We also tested 1D scalar multiplication on $\mathscr{E}$ using the Montgomery Ladder, to isolate the impact of scalar decomposition.

Table 2.1 presents our results. Compiled and ran on the same platform, our curve with the 1D Ladder (and no endomorphism) is $\approx 1.14\times$ faster than Bernstein's ladder-based `Curve25519` software [Ber06a], which is the state of the art for constant-time key exchange at the 128-bit security level. This speedup is mostly due to the faster field arithmetic afforded by working with a Mersenne prime of half the bitlength. Using the endomorphism with the 2D Bernstein chain, we are $\approx 1.23\times$ faster than Curve25519. This small improvement relative to the $\approx 1.14\times$ speedup already achieved for the 1D case is disappointing: in fact, while we halve

Table 2.1: Scalar multiplication performance, on 10^6 random inputs, for the curve of §2.5. C/assembly implementation on an Intel Core i7-3520M processor (2.89GHz, hyperthreading and Turbo Boost disabled), compiled with `gcc` v4.6.3 `-O2`. Cycles counted using the SUPERCOP toolkit [BL]. CT denotes constant-time. `xDBLADD` denotes combined `xDBL` and `xADD` with shared operands.

Multiexponentiation Addition chain	Dim.	CT	`xDBL` av.	`xDBL` s.d.	`xADD` av.	`xADD` s.d.	`xDBLADD` av.	`xDBLADD` s.d.	kCycles av.
`Ladder` [Mon87]	1	✓	1	—	—	—	253	—	159
[Ber06b]	2	✓	1	—	128	—	127	—	148
[AK14]	2	✗	1	—	1	—	179.6	6.7	133
`PRAC` [Mon87]	2	✗	0.2	0.4	113.8	11.6	73.4	11.1	109

the number of `xDBL`s, most of our savings are negated by awkward field exponentiations needed to evaluate $\psi - 1$ and normalize the images of ψ and $\psi - 1$.

The $\mathbb{Q}$-curve construction was subsequently used to develop Microsoft's FourQ software, currently the world's fastest Diffie–Hellman key exchange. The FourQ curve is in the intersection of the $d = 2$ and $d = 5$ $\mathbb{Q}$-curve families over $\mathbb{F}_{p^2}$ with the Mersenne $p = 2^{127} - 1$; of the handful of intersections over this fast $\mathbb{F}_{p^2}$, this curve luckily has a cryptographically strong subgroup! This allows 4D scalar decompositions, which speeds up scalar multiplication dramatically when used with Edwards coordinates and a fixed-window multiexponentiation.

2.6 Accelerating point counting

To apply scalar decomposition in real cryptosystems, we must find secure elliptic curves from the families above—which means computing their orders. It would be wrong to do this without exploiting their special endomorphism structures.

The task of computing $\#\mathscr{E}(\mathbb{F}_q)$ is called *point counting*. Hasse's theorem tells us that

$$\#\mathscr{E}(\mathbb{F}_q) = q + 1 - t \quad \text{where} \quad t^2 \le 4q\,.$$

Here, t is the trace of the Frobenius endomorphism π_q; all efficient point-counting algorithms are based on computing invariants of the action of Frobenius on some module associated with $\mathscr{E}$. For large p, which is the case most relevant here, the classic algorithm is the Schoof–Elkies–Atkin (SEA) algorithm [Sch85; Sch95].

GLV curves (as in Example 1) are reductions of CM curves over number fields, so we can derive the curve order from p and the CM discriminant, as in [Sch95, §4]. For GLS curves (as in Example 2), we can use the fact that

$$\#\mathscr{E}_0(\mathbb{F}_p) = p + 1 - t_0 \implies \#\mathscr{E}(\mathbb{F}_{p^2}) = (p + 1 - t_0)(p + 1 + t_0)$$

to reduce point-counting on GLS curves over $\mathbb{F}_{p^2}$ to point-counting on general elliptic curves over $\mathbb{F}_p$, where we apply SEA as usual. The $\mathbb{Q}$-curve reductions of §2.4 do not have fixed CM discriminants or subfield structures, so none of those tricks apply. But [MSS16] shows that we can still improve on SEA.

Recall that in the SEA algorithm, we detect small primes ℓ splitting in $\mathbb{Z}[\pi_{p^2}]$ by computing roots of $\Phi_\ell(j(\mathscr{E}), X)$ where Φ_ℓ is a modular polynomial; for split ℓ, we construct the polynomials $f_\ell(X)$ defining the corresponding ℓ-isogeny kernels

Table 2.2: Average time to compute $\#\mathscr{E}_{2,s}(\mathbb{F}_{p^2})$ for $1 \le s \le 100$, with $\mathscr{E}_{2,s}$ as in §2.4. Algorithms implemented in C++ using NTL 9.6.4, compiled with `gcc` 4.9.2, and ran on an Intel Xeon platform (E5520 CPU at 2.27GHz). "Powering" denotes time spent computing X^p or X^{p^2} modulo kernel polynomials.

	128-bit $p = 314159\ldots459$			255-bit $p = 314159\ldots963$		
Algorithm	av. max ℓ	Powering	Total	av. max ℓ	Powering	Total
SEA [Sch95]	164	9.11s	20.11s	352	89.73s	171.95s
New [MSS16]	62	2.62s	4.10s	160.76	22.55s	39.16s

using Elkies' algorithm, and then compute the eigenvalues of π_{p^2} on these kernels using explicit symbolic computations (dominated by computing $X^{p^2} \bmod f_\ell(X)$). The eigenvalues determine the trace $t_{\mathscr{E}}$ of π_{p^2} modulo ℓ, and after sufficiently many ℓ we can recover $t_{\mathscr{E}}$, and hence $\#\mathscr{E}(\mathbb{F}_{p^2}) = p^2 + 1 - t_{\mathscr{E}}$, using the CRT.

In [MSS16] we use ψ instead of π_{p^2}. The eigenvalues of ψ determine the eigenvalues of π_{p^2} (and hence $t_{\mathscr{E}}$, and $\#\mathscr{E}(\mathbb{F}_{p^2})$) because $\psi^2 = \pm[d]\pi_{p^2}$; and

$$\psi^2 - [r][d]\psi + [dp] = 0 \quad \text{in } \mathrm{End}(\mathscr{E}), \quad \text{where} \quad r^2 < 4p/d.$$

Compared with $t_{\mathscr{E}} = \epsilon_p(dr^2 - 2p)$, the integer r is determined by its value modulo half as many primes ℓ, and of about half the size. As a bonus, ψ is easier to evaluate than π_{p^2} on symbolic points (we only need X^p, rather than X^{p^2}, modulo $f_\ell(X)$).

Globally, the algorithm of [MSS16] has the same asymptotic complexity as SEA, but with better constants: halving the number and size of the primes ℓ implies a $\approx 4\times$ speedup, and our experiments confirm this (see Table 2.2). This speedup is welcome when searching for twist-secure curves, which requires point counting on thousands of candidate curves; when doing this, we use an early-abort variant which stops when the order mod ℓ is incompatible with a secure curve order.

3 Classical hyperelliptic cryptography

What is so special about elliptic curves? Formally, we can replace elliptic curves in cryptosytems with more general algebraic groups over $\mathbb{F}_p$: *principally polarized abelian varieties* (PPAVs). A g-dimensional PPAV is a projective algebraic variety—hence, with points corresponding to tuples of field elements—and with a group law defined by polynomial mappings. Elliptic curves are the case $g = 1$.

This chapter surveys some contributions to *hyperelliptic* curve-based cryptography. Hyperelliptic curves of genus g are the most accessible source of efficient g-dimensional PPAVs. After quickly reviewing some background in §3.1, we consider the problem of realising isogenies and endomorphisms of hyperelliptic Jacobians in §3.2, before applying this to scalar multiplication in §3.3 and point counting in §3.4. These results are all specialized to the case $g = 2$; in §3.5, we explain how to use explicit isogenies to attack discrete logarithm problem instances in genus $g = 3$. Finally, we describe a surprising new possibility for constructing trustless unknown-order groups from genus-3 curves in §3.6.

3.1 Hyperelliptic curves and cryptosystems

Jacobians of hyperelliptic curves are the most accessible examples of PPAVs. A hyperelliptic curve of genus g is defined by an affine equation

$$\mathscr{X} : y^2 = f(x) \qquad \text{with } f \text{ squarefree and } \deg(f) \in \{2g+1, 2g+2\}.$$

Divisors on $\mathscr{X}$ are formal sums of points on $\mathscr{X}$; the divisors on $\mathscr{X}$ form a group $\mathrm{Div}(\mathscr{X})$ under addition. The degree of a divisor $\sum_{P\in\mathscr{X}} n_P P$ is the sum of its coefficients, $\sum_{P\in\mathscr{X}} n_P$; the divisors of degree 0 form a subgroup $\mathrm{Div}^0(\mathscr{X}) \subset \mathrm{Div}(\mathscr{X})$. Each nonzero rational function f on $\mathscr{X}$ has an associated *principal divisor* $\mathrm{div}(f)$, which is the formal sum of its zeroes minus its poles (all counted with multiplicity). Now every principal divisor has degree 0 and $\mathrm{div}(fg) = \mathrm{div}(f) + \mathrm{div}(g)$, so $\mathrm{Prin}(\mathscr{X})$ is a subgroup of $\mathrm{Div}^0(\mathscr{X})$. The quotient $\mathrm{Div}(\mathscr{X})/\mathrm{Prin}(\mathscr{X})$ is the *Picard group* $\mathrm{Pic}(\mathscr{X})$, and the quotient $\mathrm{Div}^0(\mathscr{X})/\mathrm{Prin}(\mathscr{X})$ is the *degree-0 Picard group* $\mathrm{Pic}^0(\mathscr{X})$.

In fact $\mathrm{Pic}(\mathscr{X}) \cong \mathrm{Pic}^0(\mathscr{X}) \times \mathbb{Z}$, and $\mathrm{Pic}^0(\mathscr{X})$ is parametrized by a g-dimensional PPAV $\mathrm{Jac}(\mathscr{X})$, the *Jacobian* of $\mathscr{X}$. That is: points on $\mathrm{Jac}(\mathscr{X})$ correspond to degree-0 divisor classes on $\mathscr{X}$. We usually compute with $\mathrm{Jac}(\mathscr{X})$ using the *Mumford representation* [Mum84], where points on $\mathrm{Jac}(\mathscr{X})$ are encoded as pairs of polynomials $(a(x), b(x))$ such that $a(x)$ is monic, $\deg(b) \le \deg(a) \le g$, and $b^2 \equiv f \pmod{a}$. The group law is worked out in detail by Cantor in [Can87].

Koblitz proposed Jacobians of genus-g hyperelliptic curves as a basis for cryptosytems in 1989 [Kob89]. If $\mathscr{X}$ has genus $g \ll q$ then $\#\mathrm{Jac}(\mathscr{X}) \approx q^g$, and Pollard

rho solves the DLP in time $\widetilde{O}(q^{g/2})$; so we can trade higher genera g against proportionally smaller field primes p, potentially reducing the overall cost of cryptographic operations. But index calculus algorithms such as [Gau+07], which runs in time $\widetilde{O}(q^{2-2/g})$ for $g > 2$, have reduced the values of g for which the resulting cryptosystems offer competitive efficiency and key sizes.

Nowadays, especially after the results described in §3.5, we are mostly confined to $g = 2$ for (constructive) cryptographic applications. In that case, the *Kummer surface* $\mathrm{Kum}(\mathscr{X}) := \mathrm{Jac}(\mathscr{X})/\langle\pm 1\rangle$ is a particularly convenient object to compute with: Kummer arithmetic in genus 2 corresponds to x-only arithmetic for elliptic curves [Hud05; CF96]. Gaudry [Gau07] initiated the use of Kummer surfaces in high-speed cryptographic implementations, where they have proven competitive with elliptic curve implementations (see Chapter 5).

3.2 Realizing higher-genus isogenies with correspondences

When computing isogenies of higher-dimensional abelian varieties, the first question is how the isogenies should be represented. In my Ph.D. thesis [Smi06], I argued that (divisorial) *correspondences* are a convenient representation for isogenies and endomorphisms of Jacobians of curves.

Let $\mathscr{X}$ and $\mathscr{Y}$ be algebraic curves, and consider the product surface $\mathscr{X} \times \mathscr{Y}$; we write $\pi_1 : \mathscr{X} \times \mathscr{Y} \to \mathscr{X}$ and $\pi_2 : \mathscr{X} \times \mathscr{Y} \to \mathscr{Y}$ for the projections onto each factor. A *correspondence* is a divisor on $\mathscr{X} \times \mathscr{Y}$: that is, a formal sum of curves

$$\mathscr{R} \subset \mathscr{X} \times \mathscr{Y}\,.$$

If we suppose $\mathscr{R}$ is a prime divisor and not supported on the fibres of π_1 or π_2, then we can compose the pullback $(\pi_1|_{\mathscr{R}})^* : \mathrm{Div}(\mathscr{X}) \to \mathrm{Div}(\mathscr{R})$ with the pushforward $(\pi_2|_{\mathscr{R}})_* : \mathrm{Div}(\mathscr{R}) \to \mathrm{Div}(\mathscr{Y})$ to get an *induced homorphism* $\phi_{\mathscr{R}} : \mathrm{Div}(\mathscr{X}) \to \mathrm{Div}(\mathscr{Y})$. This homomorphism maps principal divisors onto principal divisors, and degree-0 divisor classes onto degree-0 divisor classes, so we have induced homomorphisms $\phi_{\mathscr{R}} : \mathrm{Pic}(\mathscr{X}) \to \mathrm{Pic}(\mathscr{Y})$ and $\phi_{\mathscr{R}} : \mathrm{Jac}(\mathscr{X}) \to \mathrm{Jac}(\mathscr{Y})$.[1] In fact, every homomorphism $\phi : \mathrm{Jac}(\mathscr{X}) \to \mathrm{Jac}(\mathscr{Y})$ can be represented by a correspondence on $\mathscr{X} \times \mathscr{Y}$; but this representation is never unique: we can add and subtract principal divisors and fibres of π_1 and π_2 without changing the induced homomorphism.

Suppose we want to evaluate $\phi_{\mathscr{R}}$ at a divisor class $[D]$ in $\mathrm{Pic}(\mathscr{X})$. We can always choose a representative $\sum_i P_i$ for $[D]$ such that the fibres $\pi_1^{-1}(P_i)$ intersect transversally with $\mathscr{R}$; taking that intersection yields a divisor on $\mathscr{R}$, which we then map onto $\mathscr{Y}$ via π_2. Transforming this geometric process into a fast algorithm depends on the geometry of $\mathscr{X}$ and $\mathscr{Y}$, and on the choice of correspondence.

3.3 Hyperelliptic endomorphisms and scalar multiplication

In [Smi06, Chapter 7] and [KS06], we extend endomorphism-accelerated scalar multiplication to families of hyperelliptic Jacobians over large prime fields. As in §2.4, the advantage is that we can construct one- and two-parameter families of curves whose Jacobians have efficiently computable endomorphisms over a fixed $\mathbb{F}_p$; we can certainly find cryptographically strong Jacobians in these families.

[1] Going further: the map $\mathscr{R} \to \phi_{\mathscr{R}}$ extends $\mathbb{Z}$-linearly to give a homomorphism $\mathrm{Div}(\mathscr{X} \times \mathscr{Y}) \to \mathrm{Hom}(\mathrm{Jac}(\mathscr{X}), \mathrm{Jac}(\mathscr{Y}))$, which factors through the Picard group of $\mathscr{X} \times \mathscr{Y}$ to give an isomorphism $\mathrm{Pic}(\mathscr{X} \times \mathscr{Y}) \cong \mathrm{Pic}(\mathscr{X}) \times \mathrm{Pic}(\mathscr{Y}) \times \mathrm{Hom}(\mathrm{Jac}(\mathscr{X}), \mathrm{Jac}(\mathscr{Y}))$.

We are interested in cases where the efficient endomorphisms generate rings isomorphic to orders in totally real fields. This situation is called *real multiplication*. Example 4 briefly recalls a simple one-parameter construction due to Tautz, Top, and Verberkmoes [TTV91]—itself a degeneration of a more complicated two-parameter construction due to Mestre [Mes91].

Example 4. Let p be a prime such that 5 is a square in $\mathbb{F}_p$, and let $\eta_5 = -(1+\sqrt{5})/2$ and $\bar{\eta}_5 = -(1-\sqrt{5})/2$ in $\mathbb{F}_p$ (note that $\eta_5 = \rho_5 + 1/\rho_5$ for some primitive 5-th root of unity in $\mathbb{F}_{p^2}$). Consider the one-parameter family of genus-2 curves

$$\mathscr{X}_t/\mathbb{F}_p : y^2 = D_5(x) + t \qquad \text{where} \qquad D_5(x) = x^5 - 5x^3 + 5x$$

is the degree-5 Dickson polynomial of the first kind (with parameter 1). Now

$$D_5(x_1) - D_5(x_2) = (x_1 - x_2)A_5(x_1, x_2)\bar{A}_5(x_1, x_2)$$

where

$$\begin{cases} A_5(x_1, x_2) = x_1^2 + x_2^2 + \eta_5 x_1 x_2 + \eta_5^2 - 4\,, \\ \bar{A}_5(x_1, x_2) = x_1^2 + x_2^2 + \bar{\eta}_5 x_1 x_2 + \bar{\eta}_5^2 - 4\,, \end{cases}$$

so the hypersurface $V(y_1 - y_2) \subset \mathscr{X}_t \times \mathscr{X}_t$ decomposes into three prime components: the diagonal $\Delta = (x_1 - x_2, y_1 - y_2)$, and correspondences

$$\mathscr{R}_5 := V(y_1 - y_2, A_5(x_1, x_2)) \qquad \text{and} \qquad \bar{\mathscr{R}}_5 := V(y_1 - y_2, \bar{A}_5(x_1, x_2))\,.$$

The diagonal Δ induces the identity map. There is an injection $\mathbb{Z}[(1+\sqrt{5})/2] \to \mathrm{End}(\mathrm{Jac}(\mathscr{X}_t))$ mapping $(1+\sqrt{5})/2$ to $\phi_{\mathscr{R}_t}$ and $(1-\sqrt{5})/2$ to $\phi_{\bar{\mathscr{R}}_t}$.

The endomorphisms $\phi_{\mathscr{R}_5}$ can be used to accelerate scalar multiplication on the Jacobians $\mathrm{Jac}(\mathscr{X}_t)$: their eigenvalues are very large (and the resulting scalar decompositions are suitably short) because the minimal polynomial of $(1+\sqrt{5})/2$ has very small coefficients. As with GLS and the $\mathbb{Q}$-curve construction of §2.4, the fact that $\mathscr{X}_t$ is a one-parameter family means that we can hope to find secure Jacobians over $\mathbb{F}_p$ with special p affording fast field arithmetic. It remains to show that $\phi_{\mathscr{R}_5}$ can be evaluated efficiently on points of $\mathrm{Jac}(\mathscr{X}_t)$. In [Smi06] and [KS06], we give a general method for "compiling" correspondences in the form of $\mathscr{R}_5$ to efficient programs evaluating $\phi_{\mathscr{R}_5}$ on the Mumford representation.[2]

The results of [KS06] have not yet been included in a "competitive" scalar multiplication implementation. Initially, this was due to a lack of fast point counting in genus 2, which meant computing secure instances was hard (this problem is solved in §3.4). Another problem is the explicit arithmetic of genus-2 Jacobians, which lags far behind elliptic curves when it comes to basic side-channel protections: we still do not have satisfactory constant-time group operations.

Remark 1. Example 4 is the case $n = 5$ of a more general construction in [TTV91] which, for each odd prime n, constructs a one-parameter family of curves of genus $(n-1)/2$ whose Jacobians have explicit RM by the maximal real subring of the cyclotomic field $\mathbb{Q}(\zeta_n)$. (The case $n = 3$ yields a one-parameter family of elliptic curves, but the explicit endomorphism constructed is just $[-1]$.) See [Smi06, Chapter 7] and [KS06] for more discussion and algorithms for $n \neq 5$ and the two-parameter family of [Mes91]. The cryptographic interest of $n > 5$ is limited by the impact of index calculus [Gau+07] and the results of §3.5.

[2] Similar results appeared independently in [Tak06].

3.4 Point counting with efficient real multiplication

As we saw in §2.6, efficient endomorphisms are not only useful for accelerating scalar multiplication: they can also yield significant speed-ups in point counting. The results are even more spectacular for genus-2 curves, because point-counting on these curves is already much harder than it is on elliptic curves.

Schoof's polynomial-time elliptic point-counting algorithm [Sch85] was generalized to PPAVs—including Jacobians of higher-genus curves—by Pila [Pil90]. But Pila's algorithm is only polynomial-time for PPAVs *of fixed dimension*: the implicit dependence on the dimension is badly exponential, to the point that this algorithm has never been implemented (not even in genus 2).[3] Instead, we have Schoof-type algorithms designed specifically for genus-2 Jacobians by Gaudry and Schost [GS04b; GS04a] and Pitcher [Pit09].

The record for point counting for general genus-2 curves over prime fields was set by Gaudry and Schost while searching for twist-secure Jacobians at the 128-bit security level [GS12]. Working over the Mersenne prime field $\mathbb{F}_p$ where $p = 2^{127} - 1$, they found a Jacobian whose order is 16 times a 250-bit prime and whose quadratic twist also has order 16 times a 250-bit prime. (The cofactor of 16 is required for compatibility with the high-speed Kummer arithmetic of [Gau07].) Computing one Jacobian order over this field cost around 1000 core-hours (on an Intel Xeon L5640 processor running at 2.27GHz); finding the twist-secure curve cost around 10^6 core-hours on the same processor. Looking at Table 2.2, where computing the order of a similar-sized elliptic curve on a similar processor running at the same speed cost around 20 core-seconds using the SEA algorithm, we can see that the transition from genus 1 to genus 2 takes point-counting from a routine to an extremely challenging problem.

In [GKS11], we accelerate point counting for genus-2 curves with efficient real multiplication endomorphisms, focusing on the family of Example 4. The result is a point-counting algorithm that, surprisingly enough, has *the same asymptotic complexity* as Schoof's original *elliptic* point-counting algorithm. This was used to set a new point-counting record in genus 2 (which still stands twelve years later): [GKS11] reports a kilobit Jacobian order (over $\mathbb{F}_p$ with $p = 2^{512} + 1273$) computed in 80 core-days on an Intel Core 2 running at 2.83GHz. Computing the order of one Jacobian from the family over a 128-bit field took approximately 3 core-hours.

There are two key ideas here. The first is to compute the trace of Frobenius *with respect to the real multiplication subring* $\mathbb{Z}[(-1+\sqrt{5})/2]$ instead of $\mathbb{Z}$. Thus, instead of the quartic characteristic polynomial of Frobenius, we compute a smaller quadratic polynomial—just as in Schoof's elliptic-curve algorithm. Second, we can focus on small primes ℓ that split in $\mathbb{Z}[(-1+\sqrt{5})/2]$. This gives a speed-up analogous to Elkies' speed-up for elliptic curves—but crucially, unlike SEA,

- We do not need to solve modular equations to detect ℓ splitting in the unknown endomorphism ring (corresponding to rational ℓ-isogeny kernels).
- We do not need an analogue of Elkies' algorithm[4] to reconstruct ℓ-isogeny kernels from moduli points: instead, we derive the kernel from the corre-

[3] If we fix p and consider the point-counting problem for curves over extension fields $\mathbb{F}_{p^n}$, then there is an array of efficient point counting algorithms based on explicit p-adic cohomology, derived from Kedlaya's algorithm [Ked04], which run in polynomial time with respect to n and the dimension (but exponential time with respect to $\log p$).

[4] The missing genus-2 "modular" techniques have only recently been developed by Jean Kief-

sponding ideal above ℓ in $\mathbb{Z}[\phi_{\mathscr{R}_t}] \cong \mathbb{Z}[(-1+\sqrt{5})/2]$, using the equation of $\mathscr{R}_t$ to compute the ideal of relations on the coefficients of the Mumford representation of kernel elements.

The first appearances of modular polynomials in genus 2 in [Mil15; Mar18] inspired [Bal+17],[5] which gives Atkin-style results for curves with real multiplication by $\mathbb{Z}[\sqrt{-5}]$ (including the family of Example 4). Atkin's contributions to the SEA algorithm deduce information on the trace of Frobenius from factorizations of modular polynomials. Unlike Elkies, this does not change the fundamental complexity of Schoof's algorithm, but it can make an important difference in practice. Our results could speed up computations like those of [GKS11] in the future, but they have not yet been put into practice: in any case, the complexity of the associated modular systems is so high even for small ℓ that it is unclear if or when these theoretical improvements might make a useful practical difference.

3.5 Isogenies and the DLP in genus 3

The genus-3 isogeny attack of [Smi08; Smi09] was a totally new approach to the DLP. It translates a technical and abstract result from algebraic geometry (Recillas' trigonal construction) into explicit formulas and algorithms for curves over finite fields, suitable for use by cryptographers. The impact of [Smi08] and [Smi09] is clear: these works ended all serious cryptographic research in genus 3.

Let $\mathscr{H}/\mathbb{F}_q$ be a hyperelliptic curve of genus 3; we want to solve discrete logarithms in $\mathrm{Jac}(\mathscr{H})(\mathbb{F}_q)$. The state-of-the-art solution was the index calculus of Gaudry, Thomé, Thériault, and Diem [Gau+07] running in time $\widetilde{O}(q^{4/3})$. We want to construct an explicit isogeny $\phi : \mathrm{Jac}(\mathscr{H}) \to \mathrm{Jac}(\mathscr{X})$ over $\mathbb{F}_q$, where $\mathscr{X}$ is a *non-hyperelliptic* genus-3 curve—that is, a smooth plane quartic—and use ϕ to map discrete log instances from $\mathrm{Jac}(\mathscr{H})(\mathbb{F}_q)$ into $\mathrm{Jac}(\mathscr{X})(\mathbb{F}_q)$, where they can be solved in time $\widetilde{O}(q)$ using Diem's index calculus for low-degree plane curves [Die06].

The key theoretical ingredient is Recillas' theorem [Rec74]: if $g_1 : \mathscr{C} \to \mathbb{P}^1$ is a trigonal map—that is, a map of degree 3—and $\tilde{\pi} : \widetilde{\mathscr{C}} \to \mathscr{C}$ is an unramified double cover, then there is a curve $\mathscr{X}$ of genus $g(\mathscr{C}) - 1$ such that $\mathrm{Jac}(\mathscr{X})$ is isomorphic to the Prym variety $\mathrm{Prym}(\widetilde{\mathscr{C}}/\mathscr{C})$. This $\mathscr{X}$ comes equipped with a tetragonal (degree-4) map $f : \mathscr{X} \to \mathbb{P}^1$, and conversely if we start from a tetragonal map $f : \mathscr{X} \to \mathbb{P}^1$, we can recover an unramified double cover and trigonal map $\widetilde{\mathscr{C}} \to \mathscr{C} \to \mathbb{P}^1$ such that $\mathrm{Prym}(\widetilde{\mathscr{C}}/\mathscr{C}) \cong \mathrm{Jac}(\mathscr{X})$. The isomorphism is induced by a $(3,2)$-correspondence $\mathscr{R} \subset \widetilde{\mathscr{C}} \times \mathscr{X}$, which is one of the two components of $\widetilde{\mathscr{C}} \times_{\mathbb{P}^1} \mathscr{X}$ (the other component induces the negative).

Donagi and Livné [DL99] show that this also applies when $\mathscr{C}$ is singular, provided the double cover $\widetilde{\mathscr{C}} \to \mathscr{C}$ has certain properties over the singularities—and then, if we apply Recillas' theorem to the normalization $\widetilde{\mathscr{N}}/\mathscr{N}$ of $\widetilde{\mathscr{C}}/\mathscr{C}$, then the normalization maps induce an isogeny $\mathrm{Jac}(\mathscr{X}) \cong \mathrm{Prym}(\widetilde{\mathscr{N}}/\mathscr{N}) \to \mathrm{Prym}(\widetilde{\mathscr{C}}/\mathscr{C})$. The interesting case for us is when $\widetilde{\mathscr{N}}$ is a hyperelliptic genus-3 curve $\mathscr{H}$, and $\mathscr{N} \cong \mathbb{P}^1$ is its x-line; this happens when $\tilde{\nu} : \mathscr{H} \to \widetilde{\mathscr{C}}$ is a singularization of $\mathscr{H}$ identifying disjoint pairs of Weierstrass points (and $\nu : \mathbb{P}^1 \to \mathscr{C}$ identifies their images).

fer [Kie22]. Even now, these techniques are relatively slow and complicated, mostly as a direct corollary of the (much) more complicated theory of invariants and isogenies in genus 2.

[5] [Bal+17] was the result of an intensive group project at IPAM.UCLA led by myself and Jaap Top. (This meant an unexpected return to [TTV91] for Jaap, 25 years later!)

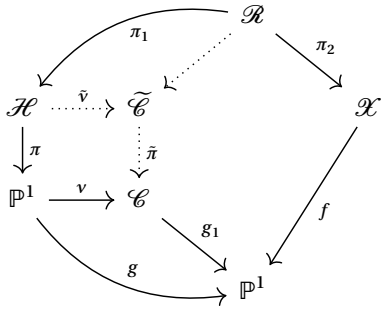

Figure 3.1: Recillas' trigonal construction in the genus-3 hyperelliptic case. The correspondence $\mathscr{R}$ induces a $(2,2,2)$-isogeny $\mathrm{Jac}(\mathscr{H}) \to \mathrm{Jac}(\mathscr{X})$, which is the $(2,2,2)$-isogeny $\mathrm{Jac}(\mathscr{H}) = \mathrm{Prym}(\mathscr{H}/\mathbb{P}^1) \to \mathrm{Prym}(\widetilde{\mathscr{C}}/\mathscr{C})$ induced by $\tilde{\nu}$ composed with Recillas' isomorphism $\mathrm{Prym}(\widetilde{\mathscr{C}}/\mathscr{C}) \cong \mathrm{Jac}(\mathscr{X})$.

Since the hyperelliptic cover $\pi : \mathscr{H} \to \mathbb{P}^1$ (i.e., the x-coordinate) ramifies precisely at the Weierstrass points, the induced cover $\tilde{\pi}$ is "of Beauville type" and amenable to the Donagi–Livné generalization of Recillas' theorem, as in Figure 3.1.

Now, let us apply this to construct a $(2,2,2)$-isogeny in genus 3, as in [Smi08]. We begin with a genus-3 hyperelliptic curve $\mathscr{H}/\mathbb{F}_q$, We want to construct an explicit isogeny from $\mathrm{Jac}(\mathscr{H})$ to the Jacobian of a non-hyperelliptic genus-3 curve $\mathscr{X}$: that is, a smooth plane quartic curve.

Choose a maximal 2-Weil-isotropic subgroup $S \subset \mathrm{Jac}(\mathscr{H})[2]$ generated by differences of Weierstrass points: $S = \langle [W_i - W_i'] \rangle_{i=1}^4$. (Not every maximal 2-Weil isotropic subgroup is generated in this way, but such subgroups will be enough for our application.) In [Smi08] we compute the composition $g = g_1 \circ \nu$ directly via linear algebra after solving a single quadraic equation, before writing down explicit equations for the codomain curve $\mathscr{X}$ *in the Mumford coordinates of* $\mathrm{Jac}(\mathscr{H})$. Equations for the correspondence $\mathscr{R}$ appear as a by-product of this representation of $\mathscr{X}$. We can easily compute the smooth plane quartic model of $\mathscr{X}$ required for Diem's index calculus using explicit Riemann–Roch [Hes02].

For this to be useful in reducing the complexity of DLP computations, we need all of the above to be done over $\mathbb{F}_q$. In fact, all of this can be done over the field of definition of S, except for two critical square roots: one to separate the two candidates for g, and one to separate the two components of $\mathscr{H} \times_{\mathbb{P}^1} \mathscr{X}$ (one of which is $\mathscr{R}$). Heuristically, we expect the first square root to be in $\mathbb{F}_q$ with probability $1/2$, and if so, then the second should also be in $\mathbb{F}_q$ with (independent) probability $1/2$; our experiments support this heuristic.

There are 105 choices for S over $\overline{\mathbb{F}}_q$, but we need an S defined over $\mathbb{F}_q$; the number available depends on the factorization of the hyperelliptic polynomial of $\mathscr{H}$, but there is always at least one defined over $\mathbb{F}_q$. For each candidate S, the attack succeeds with probability $1/4$. Summing over the possible factorization types and their probabilities, we find that the success probability of this attack on a uniformly randomly sampled hyperelliptic genus-3 curve $\mathscr{H}/\mathbb{F}_q$ is ≈ 0.1857.

This attack is only a proof of concept; it can be strengthened by using $(2,2,2)$-isogenies whose kernels are *not* generated by differences of Weierstrass points, or by (ℓ,ℓ,ℓ)-isogenies where ℓ is an odd prime, as in [LR12] and [Tia21].

3.6 Trustless unknown-order groups from genus-3 curves

As we remarked above, point-counting on hyperelliptic Jacobians of fixed genus over prime fields is polynomial-time in theory (thanks to Pila [Pil90]), but extremely hard in practice. We have already seen that for 256-bit group orders, elliptic curves take around twenty CPU-seconds with SEA, while genus-2 Jacobians take 1000 CPU-hours—five orders of magnitude greater! I am not aware of *any* serious attempt at large-characteristic point counting in genus 3 to date.

The concrete difficulty of point counting in genus 3 and beyond is not just a practical challenge: in [DGS22], we argue that for generic genus-3 curves over prime fields, point-counting can be made cryptographically infeasible. (While the idea that problems with polynomial-time solutions can be used as the basis for secure cryptosystems is obviously controversial, it goes back to the dawn of public-key cryptography [Mer78].) If this hypothesis is admitted, then we get an interesting scenario for so-called *trustless unknown-order groups*.

Many advanced cryptographic protocols run in groups of order unknown to the participants. The simplest examples of unknown-order groups are RSA groups $(\mathbb{Z}/N\mathbb{Z})^\times$ (with $N = pq$ where p and q are large primes), but establishing such a group requires a trusted third party to generate the modulus N (and to erase, or at least promise to not disclose, the prime factors p and q, from which the order of $(\mathbb{Z}/N\mathbb{Z})^\times$ can be derived). In the *trustless* setting, where no such trusted third party exists, we generally turn to ideal class groups of quadratic imaginary fields: if the participants choose a sufficiently large prime p, then the order of $\mathrm{Cl}(-p)$ is not known; the best known algorithms for computing the order are subexponential in $\log p$, so p can therefore be chosen in public without compromising security.

In [DGS22] we re-evaluate parameter sizes for secure trustless unknown-order groups in the light of Sutherland's generic order-finding algorithm [Sut07], and argue that random hyperelliptic genus-3 Jacobians might provide a competitive alternative (in terms of key sizes; it was hard to compare speeds given the lack of competitive implementations of genus-3 hyperelliptic curve arithmetic). Independent of the genus-3 proposal, [DGS22] contains a very useful result for class-group based cryptography: inspired by Bleichenbacher's Rabin signature compression algorithm, I introduced a simple method to efficiently compress class group elements to 3/4 of their previous encoded size. This improvement is particularly welcome given our new, higher estimates of parameter sizes required for secure class groups.

4 Isogeny-based cryptography

Isogeny-based cryptography (IBC) is one of the youngest areas of post-quantum cryptography. The best-known isogeny-based protocol is Jao and De Feo's key exchange [JD11; DJP14], named SIDH in [CLN16]. The key-encapsulation mechanism SIKE [Aza+17], based on SIDH, passed to the fourth round of the NIST Postquantum Standardization Project [NIST] before being spectacularly broken, along with SIDH, by Castryck and Decru [CD23], Maino, Martindale, Panny, Pope, and Wesolowski [Mai+23], and Robert [Rob23]. SIDH was itself a reaction to earlier key-exchange constructions by Couveignes [Cou06] and Rostovtsev and Stolbunov [RS06; Sto09; Sto10], which still resist these new attacks. These systems were widely considered impractical, but their more recent descendents have been more successful, and we focus on those in this chapter.

We begin with a quick tour of the theory in §4.1, before summarizing some theoretical and early practical contributions in §4.2. Going into a little more depth, we discuss the Vélusqrt algorithm in §4.4, supersingularity testing in §4.3, CSIDH implementations in §4.5, and hyperelliptic IBC in §4.6.

4.1 Theoretical foundations

An *action* of a commutative group G (written multiplicatively) on a set X is a mapping $* : G \times X \to X$ such that

$$1 * x = x \qquad \text{and} \qquad g * (h * x) = (gh) * x \quad \text{for all } g, h \in G \text{ and } x \in X\,.$$

Each choice of x in X defines a map $\varphi_x : g \mapsto g * x$ of G into X. Any commutative group action $G \times X \to X$ can be used to construct a simple key exchange analogous to Diffie–Hellman, but based on the hardness of inverting φ_x. Alice and Bob have keypairs $(g_A, x_A = g_A * x_0)$ and $(g_B, x_B = g_B * x_0)$, respectively (where x_0 is some fixed base in in X), and they derive a shared secret

$$g_A * x_B = g_A * (g_B * x_0) = g_B * (g_A * x_0) = g_B * x_A\,.$$

For example, if $X = \langle x \rangle$ is cyclic of order p and $G = (\mathbb{Z}/p\mathbb{Z})^\times$ acts on $X \setminus \{1\}$ by $g * x = x^g$, then inverting φ_x is the discrete logarithm problem (DLP) in X. But inverting φ_x for other actions may not be related to any DLP, and might resist attacks based on Shor's quantum algorithm. Indeed, inverting φ_x can be recognised as the *abelian hidden shift problem*, for which the best quantum attacks are subexponential algorithms due to Kuperberg [Kup05; Kup13] and Regev [Reg04].

Couveignes [Cou06] and—independently—Rostovtsev and Stolbunov [RS06; Sto09; Sto10] proposed using a classic group action based on the theory of Complex Multiplication [Cox13]. Let X be the set of elliptic curves $\mathscr{E}/\mathbb{F}_q$ with $\mathrm{End}(\mathscr{E}) \cong$

$\mathcal{O}$ for some quadratic order $\mathcal{O}$. If $\mathfrak{g}$ is an ideal in $\mathcal{O}$, then the finite subgroup $\mathcal{E}[\mathfrak{g}] := \bigcap_{\phi\in\mathfrak{g}} \ker\phi$ is the kernel of an isogeny $\mathcal{E} \to \mathfrak{g} * \mathcal{E} := \mathcal{E}/\mathcal{E}[\mathfrak{g}]$. Principal ideals correspond to endomorphisms, so this action induces an action of the ideal class group of $\mathcal{O}$ on X. In a sense, the resulting group-action Diffie–Hellman obfuscates Buchmann and Williams' classical class-group Diffie–Hellman [BW88; BW89].

Rostovtsev and Stolbunov advertised their cryptosystems as potential post-quantum candidates; Childs, Jao, and Soukharev led the quantum analysis using Kuperberg's algorithm with subexponential quantum isogeny evaluation [CJS14]. (See [Ber+19] for further analysis.) Isogeny-based group-actions provide the only known post-quantum *non-interactive* key exchange (NIKE), most notably with CSIDH [Cas+18]. This group action is also a fruitful source of new schemes, including CSI-FiSh [BKV19] and its many derivatives.

4.2 From theory to practice

The survey article [Smi18] investigates the theoretical foundations of commutative IBC, focusing on the obvious analogies (and the limits of the analogies) between classical and group-action-based Diffie–Hellman. For example, the classical Pollard and Baby-step giant-step attacks on the DLP in generic groups apply almost without modification to the group-action analogue of the DLP; however, Pohlig–Hellman does not extend, and neither does Shor's quantum attack.

The equivalence of the DLP and CDHP in the group setting is a long-standing open problem. We prove a remarkably simple quantum polynomial-time equivalence for their group-action analogues in [Gal+21] (assuming an error-free, idealistic version of Shor's algorithm; this result has been extended to more "realistic" quantum scenarios by Montgomery and Zhandry [MZ22]).

Couveignes' original key-exchange scheme was never implemented. Rostovtsev implemented his scheme only at very low security levels, and when we re-implemented it in [DKS18] at the 128-bit security level we found that each action took about 2000 seconds, which is clearly impractical.

Working with with Luca De Feo and Jean Kieffer, in [DKS18] we revisited Stolbunov's scheme in the hope of improving its efficiency. Our main contrbution was to show that great improvements could be made for primes ℓ where the isogeny kernels are generated by rational points: in this case, computations with modular polynomials can be replaced with a simple application of Vélu's formulæ. The problem, then, is to construct elliptic curves with orders divisible by many small primes. It is tempting to apply the CM method, but this can only produce curves whose endomorphism rings have very small class groups (which means easily brute-forceable private keyspaces here). Instead, we were reduced to exhaustive search of ordinary curves, looking for suitable group orders with point counting. Such weak curves are extremely rare, and even after many CPU-months, the best curve we could find only improved over Stolbunov by a factor of 4.

However, this work directly inspired the authors of CSIDH (*Commutative* Supersingular Isogeny Diffie–Hellman) [Cas+18] to switch from ordinary to supersingular curves over $\mathbb{F}_p$, where the curve order can be made as smooth as desired by a careful choice of the field prime p. The resulting action of the class group of $\mathbb{Z}[\sqrt{-p}]$ on the set of supersingular curves over $\mathbb{F}_p$ has the Vélu trick of [DKS18] enabled for *every* ℓ, reducing total computation time to milliseconds. CSIDH has since become the reference group action for post-quantum cryptography.

CSIDH and its derivatives work in the $\mathbb{F}_p$-subgraph of the full supersingular graph: that is, the subgraph supported on vertices defined over $\mathbb{F}_p$ (or with j-invariants in $\mathbb{F}_p$). In [CS22] we investigate a family of generalizations of the $\mathbb{F}_p$-subgraph, one for each squarefree integer d. The key is to recognise that a curve $\mathscr{E}/\mathbb{F}_{p^2}$ has its j-invariant in $\mathbb{F}_p$ precisely when $\mathscr{E}$ is isomorphic to its Galois conjugate. In [CS22] we relax the isomorphisms to d-isogenies to form the graph whose vertices are curves over $\mathbb{F}_{p^2}$ with a d-isogeny to their conjugate and whose edges are ℓ-isogenies commuting with the d-isogenies (all up to the obvious notion of isomorphism). The reader may recognise that the curves we saw in §2.4 (in the context of scalar decomposition) are a perfect source of examples. Narrowing our focus to supersingular curves, we use the theory of orientations to set up an action of the ideal class group of $\mathbb{Z}[\sqrt{-dp}]$. We thus have a natural setting for generalized variants of CSIDH (which is the case $d = 1$). We give arguments for the security of such cryptosystems, and define a broad generalization of the Delfs–Galbraith isogeny-finding algorithm [DG16] reflecting ideas from [CLG09, §7], [Eis+20], and [Arp+21].

4.3 Supersingularity

Many historic isogeny-based cryptosystems including the Charles–Goren–Lauter hash function [CLG09] and SIDH were based on the difficulty of finding paths through supersingular isogeny graphs. Now, the emergence of CSIDH and its derivatives has maintained the emphasis on supersingular curves in IBC. Efficiently identifying and generating supersingular curves is therefore critical.

Many public-key cryptographic protocols require incoming public keys to be validated to mitigate some adaptive attacks. A CSIDH public key is deemed valid if the given Montgomery coefficient specifies a supersingular elliptic curve over the prime field. In [BGS22], we survey the current supersingularity tests used for CSIDH key validation, and implement and measure two new alternative algorithms. Our implementation shows that we can determine supersingularity substantially faster, and using less memory, than the state-of-the-art.

Table 4.1: Supersingularity test comparison over $\mathbb{F}_p$. Here, n denotes the number of primes $\ell \mid (p+1)$. Experiments ran on 500 random valid (supersingular) and 500 random invalid (ordinary) curves for CSIDH-512 on an Intel i7-10610U processor running at 4.90 GHz with TurboBoost and SpeedStep disabled, compiled with `gcc` 12.1.0. Our C code used the $\mathbb{F}_p$-arithmetic library from the CTIDH package [Ban+21a]. The product tree algorithm is CTIDH's key validation.

	Asymptotic complexity		Supersingular input			Ordinary input		
			MCycles		Stack	MCycles		Stack
Algorithm	Time	Space	Av.	Med.	Max (B)	Av.	Med.	Max (B)
Random point	$O(n \log p)$	$O(1)$	63.4	62.2	2890	65.3	62.9	2890
Product tree	$O((\log n)\log p)$	$O(\log n)$	6.7	6.1	4344	1.7	1.6	3896
Sutherland	$O(\log^2 p)$	$O(1)$	35.4	35.1	2696	0.8	0.4	2696
Doliskani	$O(\log p)$	$O(1)$	4.5	4.7	3280	2.9	2.8	3264

Existing CSIDH implementations check supersingularity of a curve $\mathscr{E}/\mathbb{F}_p$ by determining the order of a random point in $\mathscr{E}(\mathbb{F}_p)$, using the fact that $\mathscr{E}$ is supersingular if and only if $\#\mathscr{E}(\mathbb{F}_p) = p+1$ (this requires the prime factorization of $p+1$,

which is known in the context of CSIDH). We can save a lot of redundant scalar multiplication, at the cost of a little more memory, by traversing a product tree for $p+1$. Sutherland's supersingularity test [Sut12] operates on curves over $\mathbb{F}_{p^2}$. This test is based on distinguishing between the 2-isogeny graph structures of supersingular and ordinary elliptic curves over $\mathbb{F}_{p^2}$. The asymptotic complexity of this algorithm is in $O(\log^2 p)$, but we find that it performs surprisingly well in practice; our adaptation to CSIDH is extremely easy to implement, and requires very little memory. It is also the only one of the four tests considered here that always returns a proven positive or negative result. We also developed a new variant of Doliskani's test [Dol18], which uses Polynomial Identity Testing to distinguish between the p-th division polynomials of supersingular and ordinary curves over $\mathbb{F}_{p^2}$. We adapted and optimized this algorithm for Montgomery models over $\mathbb{F}_p$, replacing division polynomial evaluation with a single scalar multiplication over $\mathbb{F}_{p^2}$ followed by an easy field exponentiation.

Table 4.1 presents our experimental results. We see that Sutherland's algorithm is easily the fastest when the input is ordinary/invalid, but it lags well behind the product-tree algorithm on supersingular/valid input. Our new version of Doliskani's test is the fastest on valid input, and only twice as slow as the product-tree algorithm on invalid input.

While distinguishing supersingular curves from ordinary curves is easy, sampling random supersingular curves *with unknown endomorphism ring*[1] is curiously hard. In [Boo+22], we focus on the special problem of constructing *any* supersingular curve with unknown endomorphism ring. We try a variety of approaches: "numerically" iterating to a root of the supersingular polynomial, roots of semi-evaluated modular polynomials (where the curves of §2.4 make yet another appearance), an "inverse Schoof" algorithm, the geometry of superspecial genus-2 Jacobians and Kummer surfaces, and quantum walks on isogeny graphs. While these results have some methodological interest, ultimately this project was an epic failure: none of these approaches made the slightest impact on what increasingly appears to be a fundamentally hard problem.

4.4 The Velusqrt algorithm

Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and I defined the *Vélusqrt* algorithm in [Ber+20], radically improving isogeny evaluation complexity from $O(\ell)$ to $\widetilde{O}(\sqrt{\ell})$ $\mathbb{F}_q$-operations, This is the first genuine advance in elliptic isogeny evaluation in the half-century since Vélu's algorithm appeared. Vélusqrt improves the efficiency of all IBC systems that use larger-degree isogenies, most notably CSIDH.

The central idea is that Vélu's formulæ suggest a linear sweep of the points in the kernel, and we can improve on this using a space-time tradeoff inspired by Shanks' baby-step giant-step algorithm.

Let $\mathscr{E}/\mathbb{F}_q$ be an elliptic curve with a cyclic subgroup $\mathscr{G} = \langle P \rangle$ of odd order ℓ; for convenience, we write x_s for $x([s]P)$. We want to compute the quotient isogeny $\phi : \mathscr{E} \to \mathscr{E}/\mathscr{G}$. The key task is to evaluate the kernel polynomial

$$\Psi_{\mathscr{G}}(X) = h_S(X) := \prod_{s \in S} (X - x_s) \qquad \text{where} \qquad S = \{1, 3, \dots, \ell - 4, \ell - 2\}.$$

[1] Unknown endomorphism rings are critical to the security of some isogeny-based cryptosystems: the KLPT algorithm [Koh+14] uses the endomorphism ring to efficiently solve many supersingular isogeny problems.

The classic approach to evaluating $h_S(X)$ is to apply Horner's rule, computing the sequence of x_s as $x_1 = x(P)$, $x_2 = \texttt{xDBL}(x_1)$, and then $x_{i+1} = \texttt{xADD}(x_i, x_1, x_{i-1})$ for $i \in \{2, \ldots, (\ell-3)/2\}$ for a total of $(\ell-3)/2$ curve operations and hence $O(\ell)$ $\mathbb{F}_q$-operations. (We explore a new, faster approach using mostly xDBLs in [Ban+23b], along with techniques exploiting the action of Galois when P is defined over an extension field, but these methods seem basically incompatible with Vélusqrt.)

But now, observe that if we define

$$b = \lfloor \sqrt{\ell - 1}/2 \rfloor \qquad \text{and} \qquad b' = \lfloor (\ell-1)/(4b) \rfloor ,$$

and then

$$\begin{aligned} I &= \{2b(2i+1) : 0 \le i < b'\} , \\ J &= \{2j+1 : 0 \le j < b\} , \\ K &= \{4bb'+1, \ldots, \ell-4, \ell-2\} , \end{aligned}$$

then $\#I$, $\#J$, and $\#K$ are in $O(\sqrt{\ell})$, and

$$S = (I+J) \sqcup (I-J) \sqcup K .$$

Writing $I \pm J$ for $(I+J) \sqcup (I-J)$, we have a corresponding factorization:

$$h_S(X) = h_{I \pm J}(X) h_K(X)$$

where $h_{I \pm J}(X) := \prod_{s \in I \pm J}(X - x_s)$ and $h_K(X) := \prod_{s \in K}(X - x_s)$. While $h_{I \pm J}$ has $O(\ell)$ terms, we can evaluate it (and hence h_S) using only $\widetilde{O}(\sqrt{\ell})$ $\mathbb{F}_q$-operations. The trick is to use the bijections $I \times J \to I + J$ and $I \times J \to I - J$ to express $h_{I \pm J}$ as a multivariate resultant of polynomials corresponding to I and J with degrees in $O(\sqrt{\ell})$. Concretely, let

$$E_s(X, Z) := F_0(Z, x_s) X^2 + F_1(Z, x_s) X + F_2(Z, x_s) \qquad \text{for } s \in \mathbb{Z} ,$$

where $F_0(X, Z)$, $F_1(X, Z)$, and $F_2(X, Z)$ are the biquadratic forms such that

$$(X - x(P \oplus Q))(X - x(P \ominus Q)) = X^2 + \frac{F_1(x(P), x(Q))}{F_0(x(P), x(Q))} X + \frac{F_2(x(P), x(Q))}{F_0(x(P), x(Q))}$$

for all P and Q on $\mathscr{E}$ such that $0 \notin \{P, Q, P \oplus Q, P \ominus Q\}$. Observe that if $s \equiv i \pm j \pmod{\ell}$ with $i \not\equiv \pm j \pmod{\ell}$, then x_s is a root of $E_j(X, x_i)$. Hence

$$h_{I \pm J}(X) = \Delta_{IJ}^{-1} \operatorname{Res}_Z(h_I(Z), E_J(X, Z)) \quad \text{where} \quad E_J(X, Z) := \prod_{j \in J} E_j(X, Z) ,$$

which we can use to evaluate $h_{I \pm J}(X)$ with multipoint multievaluation in only $\widetilde{O}(\sqrt{\ell})$ $\mathbb{F}_q$-operations once we have computed

$$\Delta_{IJ} := \operatorname{Res}_Z(h_I(Z), D_J(Z)) \quad \text{where} \quad D_J(Z) := \prod_{j \in J} F_0(Z, x_j) .$$

Our proof-of-concept Vélusqrt implementation already beats the classic algorithm for $\ell \ge 113$; see [Ber+20, Appendix A] for more concrete analysis and crossover points with optimized implementations. Vélusqrt is already useful for the larger primes used in the smallest CSIDH parameter set, and will be even more useful for larger parameters for higher security levels).

4.5 Constant-time CSIDH and CTIDH

Making group-action key exchanges practical is one thing; making implementations practical *and* secure is another. Some first steps towards efficient, constant-time[2] class-group actions were made by Meyer, Campos, and Reith [MCR19] and Onuki, Aikawa, Yamazaki, and Takagi [Onu+20]. At first, it looks like the actions by all of the small primes need to be made indistinguishable. For the CSIDH-512 parameter set, this would mean making 3-isogenies look like 587-isogenies, which is obviously wasteful. Instead, one of the key techniques common to all constant-time CSIDH algorithms is grouping the primes ℓ into similar-sized batches; the actions by primes in the same batch are then made indistinguishable, typically by filling in smaller ℓ-isogeny computations with dummy operations. We give some faster constant-time CSIDH actions in [Cer+19], including some *without* dummy operations, to provide some protection against simple fault attacks.

The fastest and most sophisticated constant-time CSIDH implementation is **CTIDH** [Ban+21a], which is actually faster than the original (unprotected) CSIDH algorithm. The key idea is a fundamental reimagining of the private key space. Zooming in on a single batch of primes, of norms $\ell_1 < \cdots < \ell_n$, say, most constant-time versions of CSIDH would set an exponent bound B for the batch, then sample private keys from $[-B,B]^n$ (for this batch). There are $(2B+1)^n$ keys, and computing the action of any one of them costs the same as nB ℓ_n-isogenies. In CTIDH, we instead sample exponent vectors $(e_1,\ldots,e_n)$ of bounded 1-norm from $\mathbb{Z}^n$: that is, $\sum_{i=1}^n |e_i| \leq N$ for some N. The keyspace has size $\Phi(n,N) := \sum_{k=0}^{\min(n,N)} 2^k \binom{n}{k}\binom{N}{k}$, and computing the action of any one key costs N ℓ_n-isogenies. We can generally make $\Phi(n,N) \geq (2B+1)^n$ with $N \ll nB$, which means we can save a substantial amount of work while maintaining a large keyspace (and hence a high security level).

For example, on a batch of $n = 4$ primes, taking $B = 2$ in the classic approach yields $5^4 = 625$ keys, each costing $4 \times 2 = 8$ isogenies to apply. But with the new approach, choosing arbitrary exponent vectors of 1-norm 5 yields $\Phi(4,5) = 681$ keys (a slight increase), each costing only 5 isogenies to apply—a 27% saving.

4.6 The genus-2 superspecial graph

As in Chapter 3, the natural question is: why stop at elliptic curves? In [CDS20], Wouter Castryck, Thomas Decru, and I defined a genus-2 analogue of the Charles–Goren–Lauter (CGL) hash function [CLG09] in the $(2,2)$-isogeny graph of *superspecial* principally polarized abelian surfaces (PPASes), which are isomorphic as *unpolarized* abelian varieties to products of supersingular elliptic curves.

The CGL hash uses its binary input string to drive a non-backtracking walk through the elliptic supersingular 2-isogeny graph, with the hash value being derived from the j-invariant of the end-point of the walk. Computing preimages amounts to solving the *isogeny problem* for this graph: given supersingular curves $\mathscr{E}_1$ and $\mathscr{E}_2$, efficiently compute a 2-power isogeny $\mathscr{E}_1 \to \mathscr{E}_2$.

Our superspecial hash function takes the input three bits at a time, using each three-bit chunk to choose the next step in a restricted walk in the $(2,2)$-isogeny graph of superspecial PPASes. The graph navigation is based on the modern treat-

[2] All of these algorithms are fundamentally probabilistic so we need a more subtle definition of "constant-time": the *distribution* of the run-time behaviours must be independent of any secret input.

Table 4.2: Constant-time CSIDH-512 and CSIDH-1024 implementation performance with 256-bit private keyspaces. `keygen` is key generation (one CSIDH action), `DH` is computing a shared secret (key validation and one CSIDH action). **M**, **S**, and **a** respectively denote multiplications, squarings, and additions in $\mathbb{F}_p$. Average number of millions of cycles on an Intel Xeon E3-1220 v5 (Skylake) CPU at 3GHz (except [Hut+20], ran on an Intel Core i7-7500k) with Turbo Boost disabled. For CTIDH [Ban+21a], the average was over 65 actions for each of $2^{14}-1$ private keys. Cycle counts omittted for [Onu+19] (unknown Turbo Boost) and [ACR20] (Python implementation). See [Ban+21a, §8] for more detailed measurements.

Parameters	Operation	MCycles	**M**	**S**	**a**	Reference
CSIDH-512	`keygen`	125.53	321207	116798	482311	[Ban+21a]
CSIDH-512	`keygen`	—	624000	165000	893000	[ACR20]
CSIDH-512	`DH`	129.64	330966	121787	497476	[Ban+21a]
CSIDH-512	`DH`	218.42	665876	189377	691231	[CR22]
CSIDH-512	`DH`	238.51	632444	209310	704576	[Hut+20]
CSIDH-512	`DH`	239.00	657000	210000	691000	[Cer+19]
CSIDH-512	`DH`	—	732966	243838	680801	[Onu+19]
CSIDH-512	`DH`	395.00	1054000	410000	1053000	[MCR19]
CSIDH-1024	`keygen`	469.52	287739	87944	486764	[Ban+21a]
CSIDH-1024	`keygen`	—	552000	133000	924000	[ACR20]
CSIDH-1024	`DH`	511.19	310154	99371	521400	[Ban+21a]

ment of classical Richelot isogenies [Ric37] in my Ph.D. thesis [Smi06, Chapters 8–9]. The result turns out to offer little advantage over CGL: the cost of CGL hashing is dominated by one square root in $\mathbb{F}_{p^2}$ per bit of input, while our function requires three parallel square roots in $\mathbb{F}_{p^2}$ per three bits of input, though p is half the size.

The real value of [CDS20] is in its crystallisation of problems on the superspecial graph for future research. The fundamental problem is the difficulty of solving the isogeny problem in the superspecial $(2,2)$-isogeny graph, but there are also more subtle problems on the expansion properties of the graph and the limiting behaviour of random walks, which Enric Florit and I addressed in [FS21a] and [FS21b].

In [CS20], Craig Costello and I attack the isogeny problem for superspecial PPAVs. Consider the superspecial PPASes over $\mathbb{F}_{p^2}$: there are $O(p^3)$ of them, almost all Jacobians of genus-2 curves, but there are also $O(p^2)$ elliptic products. In [CDS20], the elliptic products are an inconvenience for implementation, which we essentially ignore (because the $O(1/p)$ chance that a random walk will hit one is vanishingly small). In [CS20], random walks to elliptic products become the target: we expect to find them in $\widetilde{O}(p)$ field operations, and then we can complete the isogeny using elliptic isogenies (in only $\widetilde{O}(\sqrt{p})$ field operations). The resulting $\widetilde{O}(p)$ algorithm is exponentially faster than the usual generic random-walk-based algorithms, which would run in $\widetilde{O}(p^{3/2})$ time on the same graph. In higher dimensions, walking to PPAVs with elliptic factors lets us recursively solve in lower dimensions with an overall complexity growing linearly in the exponent (with respect to the dimension) rather than quadratically. This result suggests that higher-dimensional PPAVs are actually a surprisingly poor choice for IBC.

5 Compact public-key cryptography

We now turn to cryptographic implementations targeting constrained devices—in particular, microcontroller platforms with litle processing power and memory. In §5.1 we see that hyperelliptic genus-2 curves can give interesting practical improvements for key exchange and signatures in this domain. In §5.2 we estimate the real-world cost of the transition from curve-based to post-quantum signatures on these devices. Finally, in §5.3 we highlight some implementation results for a new code-based post-quantum signature scheme.

5.1 qDSA and muKummer

Gaudry first proposed Kummer surfaces as a setting for DLP-based cryptography in [Gau07], and Gaudry and Schost [GS12] computed parameters for a fast Kummer surface $\mathrm{Kum}(\mathscr{X})$ over $\mathbb{F}_{2^{127}-1}$ (targeting the 128-bit security level). In the sequel we write $\mathscr{X}$ for the Gaudry–Schost curve.

Bernstein, Chuengsatiansup, Lange, and Schwabe's work on high-speed key exchange in [Ber+14] showed that switching from elliptic x-lines to Kummer surfaces could improve efficiency on PC architectures. However, it was not at all obvious that Kummers could give faster key exchange on microcontroller architectures while simultaneously reducing the RAM footprint.

Joost Renes, Peter Schwabe, Lejla Batina, and I developed `μKummer` [Ren+16], a software package for high-speed, high-security key exchange and signatures, with a minimal memory footprint for use in low-end 8- and 32-bit microcotrollers. `μKummer` includes Diffie–Hellman key exchange on $\mathrm{Kum}(\mathscr{X})$ and Schnorr signatures [Sch89] on $\mathrm{Jac}(\mathscr{X})$, but with scalar multiplications computed on $\mathrm{Kum}(\mathscr{X})$. This project was based on theoretical ideas from [CCS16], which defined a genus-2 analogue of elliptic y-coordinate recovery formulæ for use with the Montgomery ladder: given a point P on $\mathrm{Jac}(\mathscr{X})$ and the Kummer images $\mathbf{x}([m]P)$ and $\mathbf{x}([m+1]P)$ in $\mathrm{Kum}(\mathscr{X})$ for some integer m, we can recover the full point $[m]P$ on $\mathrm{Jac}(\mathscr{X})$.

Our implementations targeted the ARM Cortex M0 (a typical low-end 32-bit architecture) and the AVR ATmega (an 8-bit architecture). `μKummer` was the first genus-2 hyperelliptic curve implementation on either of these architectures, so we could only compare with elliptic curve-based alternatives at the same security level: notably [Zhe14], [HS13], [WUW13], and [Dül+15]. This comparison is not superficial: all use efficient x-only arithmetic, which is the exact elliptic-curve analogue of Kummer surface arithmetic. To provide full scalar multiplication in a group, [WUW13] uses the y-coordinate recovery of [BJ02]; again, this is the elliptic-curve analogue of our methods from [CCS16].

Table 5.1 shows great results for genus-2 scalar multiplication on the Cortex M0. Compared with the then-fastest Diffie–Hellman implementation [Dül+15],

we reduce clock cycles by about 27%, while roughly halving code and stack footprints. For signature-compatible scalar multiplication on group elements, the state-of-the-art is [WUW13]; `μKummer` reduces the cycle count by a very impressive 75%, at the cost of an increase in code size and stack.

On the AVR ATmega we reduce the cycle count for Diffie–Hellman by about 32% compared with [Dül+15], again roughly halving the code size, and reducing the stack by about 80%. Jacobian scalar multiplication (for signatures) uses 71% fewer cycles than [WUW13], while increasing the stack by 25%.

Table 5.1: Scalar multiplication at the 128-bit security level on the ARM Cortex M0 platform (STM32F051R8 MCU on the STMF0Discovery board), compiled with `clang` v3.5.0, and an Arduino MEGA development board with an AVR ATmega2560 MCU, compiled with `gcc` v4.8.1. Only [WUW13] and [Ren+16] provide full signature-compatible SM; the others provide x-only SM for Diffie–Hellman. Code size for [Zhe14] includes a fixed-basepoint SM routine not used here; [HS13] does not report code size for stand-alone SM.

	ARM Cortex M0			AVR ATmega		
Source	Cycles	Stack	Code	Cycles	Stack	Code
[HS13] (Curve25519)	—	—	—	22 791 579	677 B	—
[Zhe14] (256-bit curve)	—	—	—	≈ 21 078 200	556 B	14 700 B
[Dül+15] (Curve25519)	3 589 850	548 B	7 900 B	13 900 397	494 B	17 710 B
[Ren+16] (Kum($\mathscr{X}$))	2 633 662	248 B	≈ 4 328 B	9 513 536	99 B	≈ 9 490 B
[WUW13] (NIST P-256)	≈ 10 730 000	540 B	7 168 B	≈ 34 930 000	590 B	16 112 B
[Ren+16] (Jac($\mathscr{X}$))	2 709 401	968 B	≈ 9 874 B	9 968 127	735 B	≈ 16 516 B

Looking more closely at the figures for [Ren+16] in Table 5.1, we can compare the costs of pure Kummer scalar multiplication (for Diffie–Hellman) and Jacobian scalar multiplication (for signatures) using our point recovery technique. The cost of using full group elements in the Jacobian is small in terms of speed, but large in terms of stack: this is almost entirely due to the point-recovery technique, which requires many simultaneously-live temporary variables.

This prompted Joost Renes and me to develop `qDSA` [RS17], a Schnorr signature scheme designed for Kummer varieties (including Kummer surfaces and elliptic x-lines) that does not require the full group structure of a Jacobian or an elliptic curve. We achieve this by modifying the usual verification equation in the Schnorr scheme (or, in our case, EdDSA [Ber+12]) to use relations on level-2 theta functions instead of group operations and group element equality. As a result, we retain all of the speed of [Ren+16] while losing the substantial memory overhead.

From a formal point of view the signature protocols of `μKummer` and `qDSA`, while closely related, are not the same: `μKummer` is an instantiation of the classic Schnorr scheme, while `qDSA` is based on the EdDSA variant. In particular, `μKummer` uses one full- and one half-length scalar multiplication for verification, while `qDSA` uses two full-length scalar multiplications. We did not use multiexponentation for verification: as we saw in §2.5, multiexponentiation offers an underwhelming speedup in the x-line setting, so we do not expect it to perform particularly well with Kummers—and in any case, it would have substantially increased the memory footprint of verification.

Table 5.2 gives our results for two `qDSA` instances: one based on `Curve25519`, recycling the scalar multiplication code from [Dül+15], and the other based on Kum($\mathscr{X}$), recycling the scalar multiplication code from `μKummer`.

On Cortex M0, genus-2 qDSA is significantly faster than elliptic qDSA (as should be expected), with a similar memory footprint. Comparing genus-2 qDSA with its predecessor μKummer, signing and verification are slightly slower (corresponding to the additional work in an EdDSA-based scheme over a classic Schnorr scheme), but we get a serious improvement on the memory front: the stack for `sign` resp. `verify` is reduced by about 57% resp. 43%, while code size is reduced by about 8%. (We did not compare with other signature schemes on Cortex M0, because μKummer appeared to be the first.)

Table 5.2: Signatures on the same ARM Cortex M0 and AVR ATmega platforms as Table 5.1. The SHAKE128 implementation for Cortex M0 forms 8 448 B of the code size in [RS17] and 6 938 B in [Ren+16]; there is room for some optimization here.

		ARM Cortex M0			AVR ATmega		
Scheme	Function	Cycles	Stack	Code	Cycles	Stack	Code
Ed25519 [NLD15]	`sign`	—	—	—	19 047 706	1 473 B	—
	`verify`	—	—	—	30 776 942	1 226 B	—
FourQ [Liu+20]	`sign`	—	—	—	5 174 800	1 572 B	25 354 B
	`verify`	—	—	—	11 003 800	4 957 B	33 372 B
qDSA/Curve25519 [RS17]	`sign`	3 889 116	660 B	18 443 B	14 067 995	512 B	21 347 B
	`verify`	6 793 695	788 B		25 355 140	644 B	
μKummer/Jac($\mathscr{X}$) [Ren+16]	`sign`	2 865 351	1 360 B	19 606 B	10 404 033	926 B	20 242 B
	`verify`	4 453 978	1 432 B		16 240 510	992 B	
qDSA/Kum($\mathscr{X}$) [RS17]	`sign`	2 908 215	580 B	18 064 B	10 477 347	417 B	17 880 B
	`verify`	5 694 414	808 B		20 423 937	609 B	

We found two schemes to compare with on AVR ATmega: [NLD15], based on Ed25519, and [Liu+20], based on FourQ (which uses endomorphisms to accelerate scalar multiplication). Our Curve25519-base qDSA implementation outperforms [NLD15]: `sign` resp. `verify` are more than 26% resp. 17% faster, while using 65% resp. 47% less stack. Comparing against [Liu+20], we see a clear trade-off between speed and size: FourQ is clearly faster, but qDSA on Curve25519 requires only a fraction of the stack space.

Perhaps surprisingly, the advantages of genus 2 are more pronounced on the 8-bit ATmega platform. μKummer is almost twice as fast as [NLD15] for signing and verification, while significantly reducing the stack; qDSA is roughly twice as fast for signing with only 28% of the stack, and one-and-a-half times faster for verification with just under half the stack. Comparing to [Liu+20], again we see a clear trade-off between speed and size, but this time the loss of speed is less pronounced.

5.2 Post-quantum software updates for low-end IoT devices

My IoT-focused research currently forms part of the Inria *Défi* **RIOT-FP** [RFP], which aims to provide "future-proof" high-assurance pre- and post-quantum security for RIOT OS [Bac+18; RIO], a free and open-source operating system for low-end IoT devices where Linux cannot run. In this context, I have been working not only on isogeny-based systems like [Ban+21a],[1] but also on implementation aspects of signature schemes across the whole post-quantum spectrum [Ban+22].

In [Ban+22], we make a transverse evaluation of candidate postquantum signature schemes across several embedded platforms. The novelty in [Ban+22] is in

[1]While CTIDH [Ban+21a] (see §4.5) was implemented on a PC, it was developed with a view towards microcontroller applications.

Table 5.3: Private key, public key, and signature sizes.

	Algorithm	Private Key (B)	Public Key (B)	Signature (B)
Pre-quantum	Ed25519	32	32	64
	ECDSA p256	32	32	64
Post-quantum	Falcon	1281	897	666
	Dilithium	2528	1312	2420
	LMS (RFC8554)	64	60	4756

its methodology. Cryptographers typically optimize a single scheme in a standalone application targeting a single specific architecture. Instead, we evaluate several competing schemes in the context of a real-world application—secure software updates for RIOT-OS using the SUIT protocol [Mor+21b; Mor+21a; ZS20]—and, emphasising portability, across three different platforms (Cortex-M4, RISC-V, and ESP32).

We took ECC signatures as a pre-quantum baseline. For post-quantum alternatives we took Falcon [Fou+] and Dilithium [Ava+], both based on structured lattices, together with hash-based Leighton–Micali signatures (LMS) [MCF19]. Falcon and Dilithium are both NIST standardization candidates [NIST]. Being stateful, LMS does not fit the requirements of the NIST call; but it is accepted as a high-security post-quantum scheme, and statefulness is actually quite natural in the context of certifying software package updates.

Table 5.3 compares keylengths and signature sizes for each of the schemes. The post-quantum algorithms have larger public key and signature sizes, generally by well over an order of magnitude. Compared with ECC signatures, Falcon's public keys are 28× larger and its signatures are 10.4× larger; Dilithium's public keys are 41× larger than elliptic-curve keys, and its signatures are 38× larger. LMS avoids this spectacular growth in public key sizes, with keys only 1.875× larger than elliptic-curve public keys; but its signatures are a massive 74.3× larger than elliptic-curve signatures.

To compare running times and memory requirements we ran the schemes on popular off-the-shelf IoT hardware provided by the open-access IoT-Lab testbed [Adj+15; IoT-LAB]. We chose platforms representing the landscape of modern 32-bit microcontroller architectures, including ARM Cortex-M, Espressif ESP32, and RISC-V. Table 5.4 presents our Cortex-M4 benchmarks (see [Ban+22] for similar results on ESP32 and RISC-V platforms).

Looking at Table 5.4, we see that post-quantum signing is generally slower than ECC: the fastest post-quantum algorithm is 7.94× slower than Ed25519 (Monocypher). But the tables are turned when we compare signature verification: the fastest pre-quantum algorithm is 2.65× slower than post-quantum Falcon. Efficient verification is valuable, but here it comes at the price of an increase in memory: post-quantum flash requirements can grow to over 11× the smallest pre-quantum flash sizes, and there is also a considerable increase in stack memory.

As a concrete example: suppose we want to add quantum resistance to a RIOT firmware update with SUIT for the nRF52840dk board used for our tests. Table 5.5 estimates the relative impact on the entire SUIT software update process, including the crucial aspect of network transfer costs and the memory resources required to actually apply the firmware update on the device.

Table 5.4: Pre- and post-quantum signature performance on a Nordic nRF52840 Development Kit: ARM Cortex-M4 MCU with 256 kB RAM and 1 MB flash. *Static* Dilithium is the implementation from [Ava+], modified to use (fixed) public keys pre-loaded in Flash.

		Sign			Verify			Ref.
Algorithm	Flash (B)	Time (ms)	Time (kTicks)	Stack (B)	Time (ms)	Time (kTicks)	Stack (B)	
Ed25519	5106	845	54111	1180	1953	125012	1300	[Bee17]
Ed25519	13852	17	1136	1420	40	2599	1936	[Vai+]
ECDSA	6498	294	18871	1084	313	20037	1024	[Tiny18]
Falcon	57613	1172	75020	42240	15	1004	4744	[Fou+]
Dilithium	11664	465	29788	51762	53	3407	36058	[Ava+]
Dilithium	26672	135	8655	35240	23	1510	19504	*Static*
LMS	12864	9224	590354	13212	123	7908	1580	[Flu21]

Table 5.5: Relative costs of adding quantum-resistance to RIOT firmware updates with SUIT on the Cortex M4 platform in Table 5.4. The SUIT manifest is the metadata including the signature. "No crypto" and "With crypto" refer to updates not including and including the signature library code in the update, respectively.

	Memory		SUIT	Data Transfer	
Signature in SUIT	Flash	Stack	manifest	No crypto	With crypto
Ed25519 / SHA256	52.4kB	16.3kB	483B	47kB	53kB
Falcon / SHA3-256	+120%	+18%	+224%	+1.1%	+120%
Dilithium / SHA3-256	+30%	+210%	+587%	+4.3%	+34%
LMS / SHA3-256	+34%	+1.2%	+984%	+9%	+43%

If the software update is just a small module update, or a small firmware update *without* crypto libraries, then speed and signature size are more important than flash. In these cases, Falcon has an advantage over LMS and Dilithium.

The case of small firmware updates *including* crypto libraries is more complicated, with flash playing more important role. Since we must transfer the update over a low-power network, the package size has a higher impact on energy costs. It takes 30-60s to transfer 50kB on a low-power IEEE802.15.4 radio link (depending on link quality and network load), but signature verification only varies by ±2s across all the candidates. In this case, LMS presents the best tradeoff between flash size, network transfer costs, verification time, and stack size.

For larger updates, the large network transfer costs overwhelm the other costs, minimising the advantage of one post-quantum signature over another.

5.3 Wavelet: adventures in code-based cryptography

WAVE [DST19] is a post-quantum signature scheme built from a novel trapdoor based on a hard problem in coding theory. It has the attractive feature of having relatively short signatures and fast verification, but at the cost of an *enormous* public key. This, together with the unusual feature of being based on codes over

$\mathbb{F}_3$, creates some interesting algorithmic challenges.

In [Ban+21b] we define `Wavelet`, the first full implementation of WAVE, including the first constant-time WAVE key-generation and signing algorithms.[2] Most of the features of `Wavelet` have been absorbed into the WAVE submission [Ban+23a] to the NIST post-quantum standardization signature on-ramp [NIST-sigs].

The contributions of [Ban+21b] include truncated signatures and fast signature compression, close to the information-theoretic optimum. `Wavelet` offers sub-kilobyte signatures for 128-bit classical and NIST Level-1 postquantum security: these are the smallest signatures in code-based cryptography, and are competitive with lattice-based schemes. We also define a much faster verification algorithm based on a simple transformation that allows us to skip reading almost half of the public key for any given signature—and given the size of WAVE public keys, reading them from memory is a major performance bottleneck.

Table 5.6 shows timings for our open C implementation based on high-speed bitsliced $\mathbb{F}_3$-vector arithmetic (following [Coo13]). We benchmark WAVE key generation and signing, and both classic WAVE and faster `Wavelet` verification, on an Intel Core 64-bit platform. We implemented both verifications with AVX instructions for speed, and without AVX for portability. Improving the painfully slow constant-time signing algorithm is the subject of current research.

We also tested `Wavelet` verification on an ARM Cortex M4 platform. The very large public keys (over 3MB) had to be key pre-loaded into extended flash memory and accessed via QSPI. We see that `Wavelet`'s short signatures and fast verification make it an interesting option for post-quantum signatures on microcontrollers, at least where long-term keys can be stored in a large amount of external memory.

Table 5.6: `Wavelet` performance (averaged over 100 runs) on an Intel Core i5-1135G7 processor with Turbo Boost and SpeedStep disabled, compiled with `gcc` 12.2.0 `-O3` (cycles counted with the SUPERCOP toolkit [BL]); and on a nRF52840 Development Kit with an ARM Cortex M4 MCU running at 64 MHz with 256 kB RAM, 1 MB flash, and 64 MB of external memory, compiled with GNU ARM Embedded Toolchain v11.2.0 `-O2s`. CT denotes constant-time algorithms. Key generation and signing were not implemented with AVX or for Cortex M4, and classic Wave verification was not implemented for Cortex M4.

	Intel Core		Intel Core (AVX)		ARM Cortex M4	
Operation	Time (ms)	Cycles	Time (ms)	Cycles	Time (ms)	Ticks
Key Generation	1844.1	447×10^7				
Key Generation (CT)	50199.5	566×10^8				
Sign	303.95	105×10^7				
Sign (CT)	4911.19	92×10^8				
WAVE Verification	1.67	365×10^4	1.09	300×10^4		
`Wavelet` Verification	0.37	91×10^4	0.19	76×10^4	402	13 172

[2] The proof-of-concept implementation of the WAVE trapdoor published with [DST19] at [Wave19] can generate test vectors for the trapdoor, but it cannot be used to sign messages: it does not include the required ternary hash, and its key generation and trapdoor inversion are not constant-time.

6 Looking forward

Elliptic curve cryptography is dead; long live elliptic curve cryptography.

Classical ECC has reached a high state of maturity both in theory and in implementation. While the rise of quantum computers and Shor's algorithm might suggest that there is little point working on new topics in ECC, elliptic-curve research continues to be pushed forward in IBC—but also in pairing-based cryptography and (by extension) the Zero-Knowledge community, where there are few (or no) post-quantum alternatives. The Zero-Knowledge sphere is creating fascinating new problems for curve-based cryptographers, from cycles of curves [AHG22] to implementations with arithmetization in systems like PLONK [GWC19] as a target "architecture" rather than traditional binary hardware.

While the results in §4.6 suggest that higher-dimensional supersingular PPAVs might be an inefficient setting for isogeny-based cryptosystems, the theory of isogenies in higher dimension has turned out to be the crucial ingredient in the recent attacks on SIDH and SIKE [CD23; Mai+23; Rob23]. Dartois, Leroux, Robert, and Wesolowski [Dar+23] suggest that high-dimensional isogenies are a powerful path to optimizations for elliptic isogeny-based schemes like SQISign [De +20; De +23]. Further optimizing isogeny computations, and solving cryptographic and algorithmic problems in isogeny graphs, will be an important topic for years to come—not only in IBC, but also in computational number theory.

IBC only forms a small part of the spectrum of post-quantum cryptography, and will likely only play a small part in the transition to quantum-safe cryptography. One focus of my current research is rendering a variety of post-quantum cryptosystems—not only isogeny-based, but also code-based and lattice-based—suitable for applications in constrained environments. Even in code-based cryptosystems like Wave [Ban+23a], which makes no use of elliptic curves, my experience in computational number theory has been a valuable source of optimizations and mathematical transformations.

To continue and extend this work, I proposed an Inria *Action Exploratoire* in the 2022 call on *verified* implementation of new post-quantum cryptosystems for low-end microcontrollers. Inria has a strong tradition of software verification, which has already been applied to cryptographic software, but the focus is usually on verified implementations of standardized algorithms (or standardization candidates). The AEx CACHAÇA (*Compact Asymmetric Cryptography with High Assurance for Constrained Applications*), now based at Campus Cyber in La Défense, aims to include verification techniques *from the very start* of the development of new and highly experimental cryptographic algorithms. CACHAÇA will also exploit and showcase RIOT OS as a platform for developing and deploying state-of-the-art post-quantum asymmetric cryptosystems.

Publication list

Sixteen journal articles, nineteen articles in international refereed conference proceedings, one book chapter, and five preprints. Joint publications list authors in alphabetical order[1] (except [Ren+16] and [Ban+22]).

International journal articles

[Ban+21a] G. Banegas, D. J. Bernstein, F. Campos, T. Chou, T. Lange, M. Meyer, B. Smith, and J. Sotáková. "CTIDH: faster constant-time CSIDH". In: *Transaction on Cryptographic Hardware and Embedded Systems* 2021.4 (2021), pp. 351–387. DOI: 10.46586/tches.v2021.i4.351-387. URL: https://ctidh.isogeny.org (pages 3, 21, 24, 25, 28).

[BGS22] G. Banegas, V. Gilchrist, and B. Smith. "Efficient supersingularity testing over $\mathbb{F}_p$ and CSIDH key validation". In: *Mathematical Cryptology* (2022). To appear (pages 3, 21).

[CDS20] W. Castryck, T. Decru, and B. Smith. "Hash functions from superspecial genus-2 curves using Richelot isogenies". In: *Journal of Mathematical Cryptology* 14.1 (2020), pp. 268–292 (pages 3, 24, 25).

[CRS21] J.-J. Chi-Domínguez, F. Rodríguez-Henríquez, and B. Smith. "Extending the GLS endomorphism to speed up GHS Weil descent using Magma". In: *Finite Fields and Their Applications* 75 (2021). DOI: 10.1016/j.ffa.2021.101891 (page 3).

[CS18] C. Costello and B. Smith. "Montgomery curves and their arithmetic. The case of large characteristic fields". In: *Journal of Cryptographic Engineering* 8.3 (2018), pp. 227–240. DOI: 10.1007/s13389-017-0157-6 (pages 4, 5).

[CS22] M. Chenu and B. Smith. "Higher-degree supersingular group actions". In: *Mathematical Cryptology* 1.2 (2022), pp. 85–101. eprint: https://hal.inria.fr/hal-03288075 (pages 3, 21).

[DGS22] S. Dobson, S. Galbraith, and B. Smith. "Trustless unknown-order groups". In: *Mathematical Cryptology* 1.2 (2022), pp. 25–39 (pages 2, 18).

[FS21a] E. Florit and B. Smith. "An atlas of the Richelot isogeny graph". In: *RIMS Kôkyûroku Bessatsu* (2021). To appear. URL: https://hal.inria.fr/hal-03094296 (pages 3, 25).

[1] See https://www.ams.org/profession/leaders/culture/JointResearchandItsPublicationfinal.pdf.

[Gal+09] S. D. Galbraith, J. Pujolàs, C. Ritzenthaler, and B. Smith. "Distortion maps for supersingular genus two curves". In: *Journal of Mathematical Cryptology* 3.1 (2009), pp. 1–18. DOI: 10.1515/JMC.2009.001 (page 3).

[Gal+21] S. Galbraith, L. Panny, B. Smith, and F. Vercauteren. "Quantum Equivalence of the DLP and CDHP for Group Actions". In: *Mathematical Cryptology* 1.1 (2021), pp. 40–44 (pages 2, 20).

[Ler+19] R. Lercier, C. Ritzenthaler, F. Rovetta, J. Sijsling, and B. Smith. "Distributions of traces of Frobenius for smooth plane curves over finite fields". In: *Experimental Mathematics* 28.1 (2019), pp. 39–48. DOI: 10.1080/10586458.2017.1328321 (page 3).

[MRS21] F. Morain, G. Renault, and B. Smith. "Deterministic factoring with oracles". In: *Applicable Algebra in Engineering, Communication and Computing* (2021). To appear (page 3).

[MSS16] F. Morain, C. Scribot, and B. Smith. "Computing cardinalities of $\mathbb{Q}$-curve reductions over finite fields". In: *LMS Journal of Computation and Mathematics* 19.A (2016), pp. 115–129 (pages 2, 10, 11).

[Smi09] B. Smith. "Isogenies and the discrete logarithm problem in Jacobians of genus 3 hyperelliptic curves". In: *Journal of Cryptology* 22.4 (2009), pp. 505–529. DOI: 10.1007/s00145-009-9038-1 (pages 2, 16).

[Smi11] B. Smith. "Families of explicitly isogenous Jacobians of variable-separated curves". In: *LMS Journal of Computation and Mathematics* 14 (2011), pp. 179–199 (page 3).

[Smi16] B. Smith. "The $\mathbb{Q}$-curve Construction for Endomorphism-Accelerated Elliptic Curves". In: *Journal of Cryptology* 29.4 (2016), pp. 806–832. DOI: 10.1007/s00145-015-9210-8 (pages 2, 3, 8, 9).

Articles in reviewed international conference proceedings

[Ban+22] G. Banegas, K. Zandberg, A. Herrmann, E. Baccelli, and B. Smith. "Quantum-Resistant Security for Software Updates on Low-power Networked Embedded Devices". In: *ACNS 2022: 20th International Conference on Applied Cryptography and Network Security, Rome, Italy, June 20-23, 2022. Proceedings*. 2022. eprint: https://ia.cr/2021/781 (pages 3, 28, 29, 33).

[Ban+23b] G. Banegas, V. Gilchrist, A. Le Dévéhat, and B. Smith. "Fast and Frobenius: Rational isogeny evaluation over finite fields". In: To appear. 2023 (page 23).

[Ber+20] D. J. Bernstein, L. De Feo, A. Leroux, and B. Smith. "Faster computation of isogenies of large prime degree". In: *ANTS XIV*. Ed. by S. D. Galbraith. The Open Book Series 4. 2020, pp. 39–55. DOI: 10.2140/obs.2020.4.39 (pages 3, 22, 23).

[CCS16] P. N. Chung, C. Costello, and B. Smith. "Fast, Uniform Scalar Multiplication for Genus 2 Jacobians with Fast Kummers". In: *Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers*. Ed. by R. Avanzi and H. M. Heys. Vol. 10532. Lecture Notes in Computer Science. Springer, 2016, pp. 465–481. DOI: 10.1007/978-3-319-69453-5_25 (page 26).

[Cer+19] D. Cervantes-Vázquez, M. Chenu, J.-J. Chi-Domínguez, L. De Feo, F. Rodríguez-Henríquez, and B. Smith. "Stronger and Faster Side-Channel Protections for CSIDH". In: *Progress in Cryptology - LATINCRYPT 2019 - 6th International Conference on Cryptology and Information Security in Latin America, Santiago de Chile, Chile, October 2-4, 2019, Proceedings*. Ed. by P. Schwabe and N. Thériault. Vol. 11774. Lecture Notes in Computer Science. Springer Nature. Springer, 2019, pp. 173–193. DOI: 10.1007/978-3-030-30530-7_9 (pages 24, 25).

[CHS14] C. Costello, H. Hisil, and B. Smith. "Faster Compact Diffie-Hellman: Endomorphisms on the x-line". In: *Advances in Cryptology - EUROCRYPT 2014 - 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, May 11-15, 2014. Proceedings*. Ed. by P. Q. Nguyen and E. Oswald. Vol. 8441. Lecture Notes in Computer Science. Springer Berlin Heidelberg. Springer, 2014, pp. 183–200. DOI: 10.1007/978-3-642-55220-5_11 (pages 2, 9).

[CS20] C. Costello and B. Smith. "The Supersingular Isogeny Problem in Genus 2 and Beyond". In: *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings*. Ed. by J. Ding and J.-P. Tillich. Vol. 12100. Lecture Notes in Computer Science. Springer, 2020, pp. 151–168. DOI: 10.1007/978-3-030-44223-1_9 (pages 3, 25).

[DKS18] L. De Feo, J. Kieffer, and B. Smith. "Towards Practical Key Exchange from Ordinary Isogeny Graphs". In: *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, Australia, December 2-6, 2018, Proceedings, Part III*. Ed. by T. Peyrin and S. D. Galbraith. Vol. 11274. Lecture Notes in Computer Science. Springer, Cham. Springer, 2018, pp. 365–394. DOI: 10.1007/978-3-030-03332-3_14 (pages 2, 20).

[FS21b] E. Florit and B. Smith. "Automorphisms and isogeny graphs of abelian varieties, with applications to the superspecial Richelot isogeny graph". In: *Arithmetic, geometry, cryptography and coding theory 2021*. Ed. by S. Anni, V. Karemaker, and E. Lorenzo García. Contemporary Mathematics. American Mathematical Society, 2021. URL: https://hal.inria.fr/hal-03094375 (pages 3, 25).

[GKS11] P. Gaudry, D. R. Kohel, and B. Smith. "Counting Points on Genus 2 Curves with Real Multiplication". In: *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South*

Korea, December 4-8, 2011. Proceedings. Ed. by D. H. Lee and X. Wang. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, pp. 504–519. DOI: 10.1007/978-3-642-25385-0_27 (pages 2, 15, 16).

[KS06] D. R. Kohel and B. A. Smith. "Efficiently Computable Endomorphisms for Hyperelliptic Curves". In: *Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006, Proceedings*. Ed. by F. Hess, S. Pauli, and M. E. Pohst. Vol. 4076. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. Springer, 2006, pp. 495–509. DOI: 10.1007/11792086_35 (pages 2, 13, 14).

[Ren+16] J. Renes, P. Schwabe, B. Smith, and L. Batina. "μKummer: Efficient Hyperelliptic Signatures and Key Exchange on Microcontrollers". In: *Cryptographic Hardware and Embedded Systems - CHES 2016 - 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings*. Ed. by B. Gierlichs and A. Y. Poschmann. Vol. 9813. Lecture Notes in Computer Science. Springer Berlin Heidelberg. Springer, 2016, pp. 301–320. DOI: 10.1007/978-3-662-53140-2_15 (pages 3, 26–28, 33).

[RS17] J. Renes and B. Smith. "qDSA: Small and Secure Digital Signatures with Curve-based Diffie–Hellman Key Pairs". In: *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part II*. Ed. by T. Takagi and T. Peyrin. Vol. 10625. Lecture Notes in Computer Science. Springer, Cham. Springer, 2017, pp. 273–302. DOI: 10.1007/978-3-319-70697-9_10 (pages 3, 27, 28).

[Smi08] B. Smith. "Isogenies and the Discrete Logarithm Problem in Jacobians of Genus 3 Hyperelliptic Curves". In: *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*. Ed. by N. P. Smart. Vol. 4965. Lecture Notes in Computer Science. Springer, 2008, pp. 163–180. DOI: 10.1007/978-3-540-78967-3_10 (pages 2, 16, 17).

[Smi10] B. Smith. "Families of explicit isogenies of hyperelliptic Jacobians". In: *Arithmetic, geometry, cryptography and coding theory 2009*. Ed. by D. Kohel and R. Rolland. Vol. 521. Contemporary Mathematics. American Mathematical Society, 2010, pp. 121–144 (page 3).

[Smi12] B. Smith. "Computing low-degree isogenies in genus 2 with the Dolgachev–Lehavi method". In: *Arithmetic, geometry, cryptography and coding theory*. Ed. by Y. Aubry, C. Rizenthaler, and A. Zykin. Vol. 574. Contemporary Mathematics. American Mathematical Society, 2012, pp. 159–170 (page 3).

[Smi13] B. Smith. "Families of fast elliptic curves from $\mathbb{Q}$-curves". In: *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*. Ed. by K. Sako and P. Sarkar. Vol. 8269. Lecture Notes in Computer Science. Springer, 2013, pp. 61–78. DOI: 10.1007/978-3-642-42033-7_4 (pages 2, 8, 9).

[Smi15] B. Smith. "Easy scalar decompositions for efficient scalar multiplication on elliptic curves and genus 2 Jacobians". In: *Algorithmic arithmetic, geometry, and coding theory 2014*. Ed. by S. Ballet, M. Perret, and A. Zaytsev. Vol. 637. Contemporary Mathematics. American Mathematical Society, 2015, pp. 127–141 (page 7).

[Smi18] B. Smith. "Pre- and Post-quantum Diffie–Hellman from Groups, Actions, and Isogenies". In: *Arithmetic of Finite Fields - 7th International Workshop, WAIFI 2018, Bergen, Norway, June 14-16, 2018, Revised Selected Papers*. Ed. by L. Budaghyan and F. Rodríguez-Henríquez. Vol. 11321. Lecture Notes in Computer Science. Springer, Cham. Springer, 2018, pp. 3–40. DOI: 10.1007/978-3-030-05153-2_1 (pages 2, 20).

Book chapters

[Bal+17] S. Ballentine, A. Guillevic, E. L. García, C. Martindale, M. Massierer, B. Smith, and J. Top. "Isogenies for point counting on genus two hyperelliptic curves with maximal real multiplication". In: *Algebraic Geometry for Coding Theory and Cryptography*. Ed. by E. W. Howe, K. E. Lauter, and J. L. Walker. Association for Women in Mathematics Series. Springer, Cham, 2017, pp. 63–94. DOI: 10.1007/978-3-319-63931-4 (page 16).

Preprints

[Ban+21b] G. Banegas, T. Debris-Alazard, M. Nedeljković, and B. Smith. "Wavelet: Code-based postquantum signatures with fast verification on microcontrollers". 2021. eprint: https://ia.cr/2021/1432 (pages 3, 31).

[Ban+23a] G. Banegas, K. Carrier, A. Chailloux, A. Couvreur, T. Debris-Alazard, P. Gaborit, P. Karpman, J. Loyer, R. Niederhagen, N. Sendrier, B. Smith, and J.-P. Tillich. "Wave: Supporting documentation". NIST Post-Quantum Cryptography Project Signature On-Ramp Submission. 2023 (pages 3, 31, 32).

[Boo+22] J. Booher, R. Bowden, J. Doliskani, T. B. Fouotsa, S. D. Galbraith, S. Kunzweiler, S.-P. Merz, C. Petit, B. Smith, K. E. Stange, Y. B. Ti, C. Vincent, J. F. Voloch, C. Weitkämper, and L. Zobernig. "Failing to hash into supersingular isogeny graphs". 2022. eprint: https://ia.cr/2022/518 (page 22).

[Gro+15] F. Grosshans, T. Lawson, F. Morain, and B. Smith. “Factoring Safe Semiprimes with a Single Quantum Query”. arXiv preprint arXiv:1511.04385. 2015 (page 3).

[GS06] S. D. Galbraith and B. A. Smith. “Discrete logarithms in generalized Jacobians”. arXiv preprint math/0610073. 2006 (page 3).

Ph.D. thesis

[Smi06] B. A. Smith. “Explicit endomorphisms and correspondences”. University of Sydney, 2006 (pages 2, 13, 14, 25).

Bibliography

[NIST-sigs] NIST (National Institute of Standards and Technology). *Post-Quantum Cryptography: Digital Signature Schemes*. URL: https://csrc.nist.gov/projects/pqc-dig-sig (pages 3, 31).

[NIST] NIST (National Institute of Standards and Technology). *Post-Quantum Cryptography*. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography (pages 19, 29).

[ACR20] G. Adj, J. Chi-Domínguez, and F. Rodríguez-Henríquez. *On new Vélu's formulae and their applications to CSIDH and B-SIDH constant-time implementations*. https://eprint.iacr.org/2020/1109. 2020 (page 25).

[Adj+15] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne. "FIT IoT-LAB: A Large Scale Open Experimental IoT Testbed". In: *IEEE World Forum on Internet of Things (IEEE WF-IoT)*. Dec. 2015. URL: https://hal.inria.fr/hal-01213938 (page 29).

[AHG22] D. F. Aranha, Y. E. Housni, and A. Guillevic. "A survey of elliptic curves for proof systems". In: *Designs, Codes, and Cryptography* (Special issue: Mathematics of Zero-Knowledge 2022). DOI: 10.1007/s10623-022-01135-y (page 32).

[AK14] R. Azarderakhsh and K. Karabina. "A New Double Point Multiplication Algorithm and its Application to Binary Elliptic Curves with Endomorphisms". In: *IEEE Transactions on Computers* 10.63 (2014), pp. 2614–2619. ISSN: 0018-9340. DOI: 10.1109/TC.2013.112 (pages 9, 10).

[Arp+21] S. Arpin, C. Camacho-Navarro, K. E. Lauter, J. Lim, K. Nelson, T. Scholl, and J. Sotáková. "Adventures in Supersingularland". In: *Experimental Mathematics* (2021), pp. 1–28. DOI: 10.1080/10586458.2021.1926009 (page 21).

[Ava+] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. *CRYSTALS/Dilithium*. https://pq-crystals.org/ (pages 29, 30).

[Ava+06] R. M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, and F. Vercauteren. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman & Hall/CRC, 2006 (page 7).

[Aza+17] R. Azarderakhsh, B. Koziel, M. Campagna, B. LaMacchia, C. Costello, P. Longa, L. De Feo, M. Naehrig, B. Hess, J. Renes, A. Jalali, V. Soukharev, D. Jao, and D. Urbanik. *Supersingular Isogeny Key Encapsulation*. Nov. 30, 2017. URL: http://sike.org (page 19).

[Bac+18] E. Baccelli, C. Gündoğan, O. Hahm, P. Kietzmann, M. Lenders, H. Petersen, K. Schleiser, T. C. Schmidt, and M. Wählisch. "RIOT: An Open Source Operating System for Low-End Embedded Devices in the IoT". In: *IEEE Internet of Things Journal* 5.6 (Dec. 2018), pp. 4428–4440 (pages 3, 28).

[Bee17] D. Beer. *Curve25519 and Ed25519 for low-memory systems*. Oct. 2017. URL: https://www.dlbeer.co.nz/oss/c25519.html (page 30).

[Ben05] K. Bentahar. "The Equivalence Between the DHP and DLP for Elliptic Curves Used in Practical Applications, Revisited". In: *Cryptography and Coding, 10th IMA International Conference, Cirencester, UK, December 19-21, 2005, Proceedings*. Ed. by N. P. Smart. Vol. 3796. Lecture Notes in Computer Science. Springer, 2005, pp. 376–391. ISBN: 3-540-30276-X. DOI: 10.1007/11586821_25 (page 5).

[Ber+12] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang. "High-speed high-security signatures". In: *Journal of Cryptographic Engineering* 2.2 (2012), pp. 77–89 (page 27).

[Ber+14] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and P. Schwabe. "Kummer Strikes Back: New DH Speed Records". In: *Advances in Cryptology – ASIACRYPT 2014*. Ed. by P. Sarkar and T. Iwata. Springer Berlin Heidelberg, 2014, pp. 317–337 (page 26).

[Ber+19] D. J. Bernstein, T. Lange, C. Martindale, and L. Panny. "Quantum Circuits for the CSIDH: Optimizing Quantum Evaluation of Isogenies". In: *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part II*. Ed. by Y. Ishai and V. Rijmen. Vol. 11477. Lecture Notes in Computer Science. Springer, 2019, pp. 409–441. DOI: 10.1007/978-3-030-17656-3_15 (page 20).

[Ber06a] D. J. Bernstein. "Curve25519: New Diffie-Hellman Speed Records". In: *Public Key Cryptography*. Ed. by M. Yung, Y. Dodis, A. Kiayias, and T. Malkin. Vol. 3958. LNCS. Springer, 2006, pp. 207–228. ISBN: 3-540-33851-9 (pages 8, 9).

[Ber06b] D. J. Bernstein. "Differential addition chains". Feb. 2006. URL: http://cr.yp.to/papers.html#diffchain (pages 9, 10).

[BJ02] É. Brier and M. Joye. "Weierstraß Elliptic Curves and Side-Channel Attacks". In: *Public Key Cryptography*. Ed. by D. Naccache and P. Paillier. Vol. 2274. LNCS. Springer, 2002, pp. 335–345. DOI: 10.1007/3-540-45664-3_24 (page 26).

[BKV19] W. Beullens, T. Kleinjung, and F. Vercauteren. "CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations". In: *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*. Ed. by S. D. Galbraith and S. Moriai. Vol. 11921. Lecture Notes in Computer Science. Springer, 2019, pp. 227–247. DOI: 10.1007/978-3-030-34578-5_9 (page 20).

[BL] D. J. Bernstein and T. Lange. *eBACS: ECRYPT Benchmarking of Cryptographic Systems*. URL: http://bench.cr.yp.to (pages 10, 31).

[BW88] J. Buchmann and H. C. Williams. "A key-exchange system based on imaginary quadratic fields". In: *Journal of Cryptology* 1.2 (June 1988), pp. 107–118. ISSN: 1432-1378. DOI: 10.1007/BF02351719 (page 20).

[BW89] J. A. Buchmann and H. C. Williams. "A Key Exchange System Based on Real Quadratic Fields". In: *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*. Ed. by G. Brassard. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 335–343. DOI: 10.1007/0-387-34805-0_31 (page 20).

[Can87] D. Cantor. "Computing in the Jacobian of a hyperelliptic curve". In: *Mathematics of Computation* 48 (1987), pp. 95–101 (page 12).

[Cas+18] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. "CSIDH: An Efficient Post-Quantum Commutative Group Action". In: *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*. Ed. by T. Peyrin and S. D. Galbraith. Vol. 11274. Lecture Notes in Computer Science. Springer, 2018, pp. 395–427. DOI: 10.1007/978-3-030-03332-3_15 (pages 2, 20).

[Cas91] J. W. S. Cassels. *Lectures on Elliptic Curves*. Vol. 24. London Mathematical Society Student Texts. Cambridge University Press, 1991. DOI: 10.1017/CBO9781139172530 (page 4).

[CD23] W. Castryck and T. Decru. "An Efficient Key Recovery Attack on SIDH". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by C. Hazay and M. Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 423–447. DOI: 10.1007/978-3-031-30589-4_15 (pages 19, 32).

[CF96] J. W. S. Cassels and E. V. Flynn. *Prolegomena to a middlebrow arithmetic of curves of genus* 2. Vol. 230. London Mathematical Society Lecture Note Series. Cambridge University Press, Cambridge, 1996, pp. xiv+219. ISBN: 0-521-48370-0. DOI: 10.1017/CBO9780511526084 (page 13).

[CJS14] A. Childs, D. Jao, and V. Soukharev. "Constructing elliptic curve isogenies in quantum subexponential time". In: *Journal of Mathematical Cryptology* 8.1 (2014), pp. 1–29 (page 20).

[CLG09] D. X. Charles, K. E. Lauter, and E. Z. Goren. "Cryptographic hash functions from expander graphs". In: *Journal of Cryptology* 22.1 (2009), pp. 93–113 (pages 21, 24).

[CLN16] C. Costello, P. Longa, and M. Naehrig. "Efficient Algorithms for Supersingular Isogeny Diffie-Hellman". In: *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I*. Ed. by M. Robshaw and J. Katz. Vol. 9814. Lecture Notes in Computer Science. Springer, 2016, pp. 572–601. DOI: 10.1007/978-3-662-53018-4_21 (page 19).

[Coo13] K. Coolsaet. "Fast vector arithmetic over $\mathbb{F}_3$". eng. In: *Bulletin of the Belgian Mathematical Society – Simon Stevin* 20.2 (2013), pp. 329–344. ISSN: 1370-1444. URL: http://projecteuclid.org/euclid.bbms/1369316548 (page 31).

[Cou06] J. M. Couveignes. "Hard Homogeneous Spaces". In: *IACR Cryptology ePrint Archive* 2006 (2006), p. 291. URL: http://eprint.iacr.org/2006/291 (page 19).

[Cox13] D. A. Cox. *Primes of the Form $x^2 + ny^2$: Fermat, Class Field Theory, and Complex Multiplication*. 2nd. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. John Wiley and Sons, 2013. DOI: 10.1002/9781118400722 (page 19).

[CR22] J.-J. Chi-Domínguez and F. Rodríguez-Henríquez. "Optimal strategies for CSIDH". In: *Advances in Mathematics of Communication* 16 (2 2022), pp. 383–411. DOI: 10.3934/amc.2020116 (page 25).

[Dar+23] P. Dartois, A. Leroux, D. Robert, and B. Wesolowski. *SQISignHD: New Dimensions in Cryptography*. Cryptology ePrint Archive, Paper 2023/436. https://eprint.iacr.org/2023/436. 2023. URL: https://eprint.iacr.org/2023/436 (page 32).

[De +20] L. De Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski. "SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies". In: *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*. Ed. by S. Moriai and H. Wang. Vol. 12491. Lecture Notes in Computer Science. Springer, 2020, pp. 64–93. DOI: 10.1007/978-3-030-64837-4_3 (page 32).

[De +23] L. De Feo, A. Leroux, P. Longa, and B. Wesolowski. "New Algorithms for the Deuring Correspondence - Towards Practical and Secure SQISign Signatures". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by C. Hazay and M. Stam. Vol. 14008.

Lecture Notes in Computer Science. Springer, 2023, pp. 659–690. DOI: 10.1007/978-3-031-30589-4_23 (page 32).

[DG16] C. Delfs and S. D. Galbraith. "Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$". In: *Designs, Codes and Cryptography* 78.2 (2016), pp. 425–440. DOI: 10.1007/s10623-014-0010-1 (page 21).

[Die06] C. Diem. "An index calculus algorithm for plane curves of small degree". In: *Algorithmic Number Theory: ANTS-VII*. Ed. by F. Hess, S. Pauli, and M. Pohst. Vol. 4076. LNCS. 2006, pp. 543–557 (page 16).

[DJP14] L. De Feo, D. Jao, and J. Plût. "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies". In: *Journal of Mathematical Cryptology* 8.3 (2014), pp. 209–247 (page 19).

[DL99] R. Donagi and R. Livné. "The arithmetic-geometric mean and isogenies for curves of higher genus". In: *Annali della Scuola Normale Superiore di Pisa, Classe di Scienze* 28.2 (1999), pp. 323–339 (page 16).

[Dol18] J. Doliskani. "On division polynomial PIT and supersingularity". In: *Appl. Algebra Eng. Commun. Comput.* 29.5 (2018), pp. 393–407. DOI: 10.1007/s00200-018-0349-z (page 22).

[DST19] T. Debris-Alazard, N. Sendrier, and J. Tillich. "Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes". In: *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part I*. Ed. by S. D. Galbraith and S. Moriai. Vol. 11921. Lecture Notes in Computer Science. Springer, 2019, pp. 21–51. DOI: 10.1007/978-3-030-34578-5_2 (pages 30, 31).

[Dül+15] M. Düll, B. Haase, G. Hinterwälder, M. Hutter, C. Paar, A. H. Sánchez, and P. Schwabe. "High-speed Curve25519 on 8-bit, 16-bit and 32-bit microcontrollers". In: *Design, Codes and Cryptography* 77.2 (2015) (pages 26, 27).

[Eis+20] K. Eisenträger, S. Hallgren, C. Leonardi, T. Morrison, and J. Park. "Computing endomorphism rings of supersingular elliptic curves and connections to path-finding in isogeny graphs". In: *Fourteenth Algorithmic Number Theory Symposium*. Ed. by S. D. Galbraith. Vol. 4. Open book series. Mathematical Sciences Publishers, 2020, pp. 215–232. DOI: https://doi.org/10.2140/obs.2020.4.215 (page 21).

[Flu21] S. Fluhrer. *LMS Hash-Based Signature Implementation*. https://github.com/cisco/hash-sigs/. 2021 (page 30).

[Fou+] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. *Falcon: Fast-Fourier lattice-based compact signatures over NTRU*. https://falcon-sign.info (pages 29, 30).

[Fou+08] P.-A. Fouque, R. Lercier, D. Réal, and F. Valette. "Fault Attack on Elliptic Curve Montgomery Ladder Implementation". In: *FDTC*. Ed. by L. Breveglieri, S. Gueron, I. Koren, D. Naccache, and J.-P. Seifert. IEEE Computer Society, 2008, pp. 92–98. ISBN: 978-0-7695-3314-8 (page 8).

[Gal12] S. D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012. DOI: 10.1017/CBO9781139012843 (pages 4, 7).

[Gau+07] P. Gaudry, E. Thomé, N. Thériault, and C. Diem. "A double large prime variation for small genus hyperelliptic index calculus". In: *Mathematics of Computation* 76 (2007), pp. 475–492 (pages 13, 14, 16).

[Gau07] P. Gaudry. "Fast genus 2 arithmetic based on Theta functions". In: *Journal of Mathematical Cryptology* 1 (2007), pp. 243–265 (pages 13, 15, 26).

[GLS11] S. D. Galbraith, X. Lin, and M. Scott. "Endomorphisms for Faster Elliptic Curve Cryptography on a Large Class of Curves". In: *J. Cryptology* 24.3 (2011), pp. 446–469 (page 8).

[GLV01] R. P. Gallant, R. J. Lambert, and S. A. Vanstone. "Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms". In: *CRYPTO*. Ed. by J. Kilian. Vol. 2139. LNCS. Springer, 2001, pp. 190–200. ISBN: 3-540-42456-3. DOI: 10.1007/3-540-44647-8_11 (pages 7, 8).

[GS04a] P. Gaudry and É. Schost. "A Low-Memory Parallel Version of Matsuo, Chao, and Tsujii's Algorithm". In: *Algorithmic Number Theory, 6th International Symposium, ANTS-VI, Burlington, VT, USA, June 13-18, 2004, Proceedings*. Ed. by D. A. Buell. Vol. 3076. Lecture Notes in Computer Science. Springer, 2004, pp. 208–222. DOI: 10.1007/978-3-540-24847-7_15. URL: https://doi.org/10.1007/978-3-540-24847-7_15 (page 15).

[GS04b] P. Gaudry and É. Schost. "Construction of Secure Random Curves of Genus 2 over Prime Fields". In: *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*. Ed. by C. Cachin and J. Camenisch. Vol. 3027. Lecture Notes in Computer Science. Springer, 2004, pp. 239–256. DOI: 10.1007/978-3-540-24676-3_15. URL: https://doi.org/10.1007/978-3-540-24676-3_15 (page 15).

[GS12] P. Gaudry and É. Schost. "Genus 2 point counting over prime fields". In: *Journal of Symbolic Computation* 47.4 (2012). Special Issue for Joachim von zur Gathen at 60, pp. 368–400. ISSN: 0747-7171. DOI: https://doi.org/10.1016/j.jsc.2011.09.003 (pages 15, 26).

[GWC19] A. Gabizon, Z. J. Williamson, and O. Ciobotaru. "PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge". In: *IACR Cryptol. ePrint Arch.* (2019), p. 953. URL: https://eprint.iacr.org/2019/953 (page 32).

[Has97] Y. Hasegawa. "Q-curves over quadratic fields". In: *Manuscripta Mathematica* 94.1 (1997), pp. 347–364. ISSN: 1432-1785. DOI: 10.1007/BF02677859. URL: http://dx.doi.org/10.1007/BF02677859 (page 9).

[Hes02] F. Hess. "Computing Riemann–Roch spaces in algebraic function fields and related topics". In: *Journal of Symbolic Computation* 33.4 (2002), pp. 425–445 (page 17).

[HS13] M. Hutter and P. Schwabe. "NaCl on 8-bit AVR Microcontrollers". In: *Progress in Cryptology – AFRICACRYPT 2013*. Ed. by A. Youssef and A. Nitaj. Vol. 7918. LNCS. Springer, 2013, pp. 156–172 (pages 26, 27).

[Hud05] R. W. H. T. Hudson. *Kummer's quartic surface*. Cambridge University Press, 1905 (page 13).

[Hut+20] A. Hutchinson, J. T. LeGrow, B. Koziel, and R. Azarderakhsh. "Further Optimizations of CSIDH: A Systematic Approach to Efficient Strategies, Permutations, and Bound Vectors". In: *Applied Cryptography and Network Security - 18th International Conference, ACNS 2020, Rome, Italy, October 19-22, 2020, Proceedings, Part I*. Ed. by M. Conti, J. Zhou, E. Casalicchio, and A. Spognardi. Vol. 12146. Lecture Notes in Computer Science. Springer, 2020, pp. 481–501. DOI: 10.1007/978-3-030-57808-4_24 (page 25).

[IoT-LAB] *IoT-LAB: Very large scale open wireless sensor network testbed*. URL: https://www.iot-lab.info/ (visited on 07/19/2023) (page 29).

[JD11] D. Jao and L. De Feo. "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies". In: *Post-Quantum Cryptography*. Ed. by B.-Y. Yang. Vol. 7071. Lecture Notes in Computer Science. Taipei, Taiwan: Springer Berlin / Heidelberg, 2011. Chap. 2, pp. 19–34. ISBN: 978-3-642-25404-8. DOI: 10.1007/978-3-642-25405-5_2 (page 19).

[Ked04] K. S. Kedlaya. "Computing Zeta Functions via p-Adic Cohomology". In: *Algorithmic Number Theory, 6th International Symposium, ANTS-VI, Burlington, VT, USA, June 13-18, 2004, Proceedings*. Ed. by D. A. Buell. Vol. 3076. Lecture Notes in Computer Science. Springer, 2004, pp. 1–17. DOI: 10.1007/978-3-540-24847-7_1 (page 15).

[Kie22] J. Kieffer. "Counting points on abelian surfaces over finite fields with Elkies's method". In: *CoRR* abs/2203.02009 (2022). DOI: 10.48550/arXiv.2203.02009. arXiv: 2203.02009 (page 16).

[Knu97] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Third. Boston: Addison-Wesley, 1997 (page 7).

[Kob87] N. Koblitz. "Elliptic curve cryptosystems". In: *Math. Comp.* 48.177 (1987), pp. 203–209 (page 1).

[Kob89] N. Koblitz. "Hyperelliptic Cryptosystems". In: *Journal of Cryptology* 1.3 (1989), pp. 139–150. DOI: 10.1007/BF02252872 (page 12).

[Koh+14] D. R. Kohel, K. Lauter, C. Petit, and J.-P. Tignol. "On the quaternion ℓ-isogeny path problem". In: *LMS Journal of Computation and Mathematics* 17.A (2014), pp. 418–432. DOI: 10.1112/S1461157014000151 (page 22).

[Koh96] D. R. Kohel. “Endomorphism rings of elliptic curves over finite fields”. Ph.D. thesis. University of California at Berkeley, 1996 (page 6).

[Kup05] G. Kuperberg. “A subexponential-time quantum algorithm for the dihedral hidden subgroup problem”. In: *SIAM Journal of Computing* 35.1 (2005), pp. 170–188. eprint: quant-ph/0302112 (page 19).

[Kup13] G. Kuperberg. “Another Subexponential-time Quantum Algorithm for the Dihedral Hidden Subgroup Problem”. In: *8th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2013)*. Ed. by S. Severini and F. Brandao. Vol. 22. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013, pp. 20–34. DOI: 10.4230/LIPIcs.TQC.2013.20 (page 19).

[Liu+20] Z. Liu, P. Longa, G. C. C. F. Pereira, O. Reparaz, and H. Seo. “Four$\mathbb{Q}$ on Embedded Devices with Strong Countermeasures Against Side-Channel Attacks”. In: *IEEE Transactions on Dependable and Secure Computing* 17.3 (2020), pp. 536–549. DOI: 10.1109/TDSC.2018.2799844 (page 28).

[LR12] D. Lubicz and D. Robert. “Computing isogenies between abelian varieties”. In: *Compositio Mathematica* 148.5 (2012), pp. 1483–1515. DOI: 10.1112/S0010437X12000243 (page 17).

[Mai+23] L. Maino, C. Martindale, L. Panny, G. Pope, and B. Wesolowski. “A Direct Key Recovery Attack on SIDH”. In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by C. Hazay and M. Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 448–471. DOI: 10.1007/978-3-031-30589-4_16 (pages 19, 32).

[Mar18] C. Martindale. “Isogeny Graphs, Modular Polynomials, and Applications”. PhD thesis. Universiteit Leiden, June 2018 (page 16).

[Mau94] U. M. Maurer. “Towards the Equivalence of Breaking the Diffie–Hellman Protocol and Computing Discrete Algorithms”. In: *Advances in Cryptology - CRYPTO ’94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*. Ed. by Y. Desmedt. Vol. 839. Lecture Notes in Computer Science. Springer, 1994, pp. 271–281. DOI: 10.1007/3-540-48658-5 (page 5).

[MCF19] D. McGrew, M. Curcio, and S. Fluhrer. *Leighton–Micali Hash-Based Signatures*. RFC 8554. 2019. URL: https://datatracker.ietf.org/doc/html/rfc8554 (page 29).

[MCR19] M. Meyer, F. Campos, and S. Reith. “On Lions and Elligators: An Efficient Constant-Time Implementation of CSIDH”. In: *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers*. Ed. by J. Ding and R. Steinwandt. Vol. 11505. Lecture Notes

in Computer Science. Springer, 2019, pp. 307–325. DOI: 10.1007/978-3-030-25510-7_17 (pages 24, 25).

[Mer78] R. C. Merkle. "Secure communications over insecure channels". In: *Communications of the ACM* 24.4 (1978), pp. 294–299 (page 18).

[Mes91] J.-F. Mestre. "Familles de courbes hyperelliptiques à multiplications réelles". In: *Arithmetic algebraic geometry (Texel, 1989)*. Vol. 89. Progress in Mathematics. Birkhäuser Boston, 1991 (page 14).

[Mil15] E. Milio. "A quasi-linear time algorithm for computing modular polynomials in dimension 2". In: *LMS Journal of Computation and Mathematics* 18.1 (2015), pp. 603–632. DOI: 10.1112/S1461157015000170 (page 16).

[Mil85] V. S. Miller. "Use of Elliptic Curves in Cryptography". In: *CRYPTO*. Ed. by H. C. Williams. Vol. 218. LNCS. Springer, 1985, pp. 417–426. ISBN: 3-540-16463-4 (page 1).

[Mon87] P. L. Montgomery. "Speeding the Pollard and elliptic curve methods of factorization". In: *Math. Comp.* 48.177 (1987), pp. 243–264 (pages 5, 9, 10).

[Mor+21a] B. Moran, H. Tschofenig, H. Birkholz, and K. Zandberg. *A CBOR-based Serialization Format for the Software Updates for Internet of Things (SUIT) Manifest*. Internet-Draft draft-ietf-suit-manifest-12. Work in Progress. Internet Engineering Task Force, Feb. 2021. URL: https://datatracker.ietf.org/doc/html/draft-ietf-suit-manifest-12 (page 29).

[Mor+21b] B. Moran, H. Tschofenig, D. Brown, and M. Meriac. *A Firmware Update Architecture for Internet of Things*. RFC 9019. Apr. 2021. DOI: 10.17487/RFC9019. URL: https://rfc-editor.org/rfc/rfc9019.txt (page 29).

[MSV04] A. Muzereau, N. P. Smart, and F. Vercauteren. "The Equivalence between the DHP and DLP for Elliptic Curves Used in Practical Applications". In: *LMS Journal of Computation and Mathematics* 7 (2004), pp. 50–72. DOI: 10.1112/S1461157000001042 (page 5).

[Mum84] D. Mumford. *Tata lectures on Theta II*. Vol. 43. Progress in Mathematics. Birkhäuser Boston, 1984 (page 12).

[MW99] U. M. Maurer and S. Wolf. "The Relationship Between Breaking the Diffie–Hellman Protocol and Computing Discrete Logarithms". In: *SIAM J. Comput.* 28.5 (1999), pp. 1689–1721. DOI: 10.1137/S0097539796302749 (page 5).

[MZ22] H. Montgomery and M. Zhandry. "Full Quantum Equivalence of Group Action DLog and CDH, and More". In: *Advances in Cryptology - ASIACRYPT 2022 - 28th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, December 5-9, 2022, Proceedings, Part I*. Ed. by S. Agrawal and D. Lin. Vol. 13791. Lecture Notes in Computer Science. Springer, 2022, pp. 3–32. DOI: 10.1007/978-3-031-22963-3_1 (page 20).

[NLD15] E. Nascimento, J. López, and R. Dahab. "Efficient and Secure Elliptic Curve Cryptography for 8-bit AVR Microcontrollers". In: *Security, Privacy, and Applied Cryptography Engineering*. Ed. by R. S. Chakraborty, P. Schwabe, and J. Solworth. Vol. 9354. LNCS. Springer, 2015, pp. 289–309 (page 28).

[Onu+19] H. Onuki, Y. Aikawa, T. Yamazaki, and T. Takagi. "(Short Paper) A Faster Constant-Time Algorithm of CSIDH Keeping Two Points". In: *Advances in Information and Computer Security - 14th International Workshop on Security, IWSEC 2019, Tokyo, Japan, August 28-30, 2019, Proceedings*. Ed. by N. Attrapadung and T. Yagi. Vol. 11689. Lecture Notes in Computer Science. https://eprint.iacr.org/2019/353. Springer, 2019, pp. 23–33. DOI: 10.1007/978-3-030-26834-3_2 (page 25).

[Onu+20] H. Onuki, Y. Aikawa, T. Yamazaki, and T. Takagi. "A Constant-Time Algorithm of CSIDH Keeping Two Points". In: *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* 103-A.10 (2020), pp. 1174–1182. DOI: 10.1587/transfun.2019DMP0008 (page 24).

[Pil90] J. Pila. "Frobenius maps of abelian varieties and finding roots of unity in finite fields". In: *Mathematics of Computation* 55.192 (1990), pp. 745–763 (pages 15, 18).

[Pit09] N. L. Pitcher. "Efficient point-counting on genus-2 hyperelliptic curves". Ph.D. thesis. University of Illinois at Chicago, 2009 (page 15).

[Rec74] S. Recillas. "Jacobians of curves with g_4^1's are the Pryms of trigonal curves". In: *Boletín de la Sociedad Matemática Mexicana*. 2nd ser. 19.1 (1974), pp. 9–13 (page 16).

[Reg04] O. Regev. *A Subexponential Time Algorithm for the Dihedral Hidden Subgroup Problem with Polynomial Space*. arXiv:quant-ph/0406151. June 2004. URL: http://arxiv.org/abs/quant-ph/0406151 (page 19).

[RFP] *RIOT-fp: Reconcile IoT and Future-Proof Security*. URL: https://www.inria.fr/fr/riot-fp (pages 3, 28).

[Ric37] F. J. Richelot. "De transformatione integralium Abelianorum primi ordinis commentatio". In: *Journal für die reine und angewandte Mathematik* 16 (1837), pp. 285–341 (page 25).

[RIO] RIOT OS developers. *RIOT OS*. https://riot-os.org (pages 3, 28).

[Rob23] D. Robert. "Breaking SIDH in Polynomial Time". In: *Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V*. Ed. by C. Hazay and M. Stam. Vol. 14008. Lecture Notes in Computer Science. Springer, 2023, pp. 472–503. DOI: 10.1007/978-3-031-30589-4_17 (pages 19, 32).

[RS06] A. Rostovtsev and A. Stolbunov. "Public-Key Cryptosystem Based on Isogenies". In: *IACR Cryptology ePrint Archive* 2006 (2006), p. 145. URL: http://eprint.iacr.org/2006/145 (page 19).

[Sch85] R. Schoof. “Elliptic curves over finite fields and the computation of square roots mod p”. In: *Mathematics of computation* 44.170 (1985), pp. 483–494 (pages 10, 15).

[Sch89] C.-P. Schnorr. “Efficient Identification and Signatures for Smart Cards”. In: *Advances in Cryptology - CRYPTO ’89*. Ed. by G. Brassard. Vol. 435. Lecture Notes in Computer Science. Springer, 1989, pp. 239–252 (page 26).

[Sch95] R. Schoof. “Counting points on elliptic curves over finite fields”. In: *J. Théor. Nombres Bordeaux* 7.1 (1995), pp. 219–254 (pages 10, 11).

[Sho97] P. W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM J. Comput.* 26.5 (1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172 (page 1).

[Sil09] J. H. Silverman. *The Arithmetic of Elliptic Curves*. 2nd ed. Graduate Texts in Mathematics 106. Dordrecht: Springer, 2009. DOI: 10.1007/978-0-387-09494-6 (pages 4, 6).

[Sto09] A. Stolbunov. “Reductionist Security Arguments for Public-Key Cryptographic Schemes Based on Group Action”. In: *Norsk informasjonssikkerhetskonferanse (NISK)*. Ed. by S. F. Mjølsnes. 2009 (page 19).

[Sto10] A. Stolbunov. “Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves”. In: *Adv. Math. Commun.* 4.2 (2010) (page 19).

[Str64] E. G. Straus. “Addition chains of vectors”. In: *Amer. Math. Monthly* 71 (1964), pp. 806–808 (page 7).

[Sut07] A. V. Sutherland. “Order computations in generic groups”. Ph.D. thesis. Massachusetts Institute of Technology, June 2007 (page 18).

[Sut12] A. V. Sutherland. “Identifying supersingular elliptic curves”. In: *LMS Journal of Computation and Mathematics* 15 (2012), pp. 317–325. DOI: 10.1112/S1461157012001106 (page 22).

[Tak06] K. Takashima. “A new type of fast endomorphisms on Jacobians of hyperelliptic curves and their cryptographic application”. In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E89-A.1 (2006), pp. 124–133 (page 14).

[Tia21] S. Tian. “Translating the Discrete Logarithm Problem on Jacobians of Genus 3 Hyperelliptic Curves with (ℓ,ℓ,ℓ)-isogenies”. In: *Journal of Cryptology* 34.32 (June 2021). DOI: 10.1007/s00145-021-09401-3 (page 17).

[Tiny18] *TinyCrypt Cryptographic Library*. 2018. URL: https://github.com/01org/tinycrypt (page 30).

[TTV91] W. Tautz, J. Top, and A. Verberkmoes. “Explicit hyperelliptic curves with real multiplication and permutation polynomials”. In: *Canadian Journal of Mathematics* 43.5 (1991), pp. 1055–1064 (pages 14, 16).

[Vai+] L. Vaillant et al. *Monocypher*. URL: https://monocypher.org/ (visited on 07/05/2018) (page 30).

[Vél71] J. Vélu. “Isogénies entre courbes elliptiques”. In: *Comptes Rendus Hebdomadaires des Séances de l’Académie des Sciences, Série A* 273 (July 1971), pp. 238–241 (page 6).

[Wave19] *Wave: a code-based digital signature scheme*. 2019. URL: https://wave.inria.fr/ (page 31).

[WUW13] E. Wenger, T. Unterluggauer, and M. Werner. “8/16/32 Shades of Elliptic Curve Cryptography on Embedded Processors”. In: *Progress in Cryptology – INDOCRYPT 2013*. Ed. by G. Paul and S. Vaudenay. Vol. 8250. LNCS. Springer, 2013, pp. 244–261 (pages 26, 27).

[Zhe14] J. G. Zhe Liu Erich Wenger. “MoTE-ECC: Energy-Scalable Elliptic Curve Cryptography for Wireless Sensor Networks”. In: *Applied Cryptography and Network Security*. Ed. by I. Boureanui, P. Owesarski, and S. Vaudenay. Vol. 8479. LNCS. Springer, 2014, pp. 361–379 (pages 26, 27).

[ZS20] K. Zandberg and K. Schleiser. *SUIT Reference Implementation*. RIOT. RIOT, 2020. URL: http://api.riot-os.org/group__sys__suit.html (page 29).