Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies

David Jao[1] and Luca De Feo[2]

[1] Department of Combinatorics and Optimization
University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada
djao@math.uwaterloo.ca

[2] Laboratoire PRiSM
Université de Versailles, 78035 Versailles, France
http://www.prism.uvsq.fr/~dfl

Abstract. We present new candidates for quantum-resistant public-key cryptosystems based on the conjectured difficulty of finding isogenies between supersingular elliptic curves. The main technical idea in our scheme is that we transmit the images of torsion bases under the isogeny in order to allow the two parties to arrive at a common shared key despite the noncommutativity of the endomorphism ring. Our work is motivated by the recent development of a subexponential-time quantum algorithm for constructing isogenies between ordinary elliptic curves. In the supersingular case, by contrast, the fastest known quantum attack remains exponential, since the noncommutativity of the endomorphism ring means that the approach used in the ordinary case does not apply. We give a precise formulation of the necessary computational assumption along with a discussion of its validity, and prove the security of our protocols under this assumption. In addition, we present implementation results showing that our protocols are multiple orders of magnitude faster than previous isogeny-based cryptosystems over ordinary curves.

Keywords: elliptic curves, isogenies, quantum-resistant public-key cryptosystems

1 Introduction

The Diffie-Hellman scheme is a fundamental protocol for public-key exchange between two parties. Its original definition over finite fields is based on the hardness of computing the map $g, g^a, g^b \mapsto g^{ab}$ for $g \in \mathbb{F}_p^*$, while its elliptic curve analogue depends on the difficulty of computing $P, aP, bP \mapsto abP$ for points P on an elliptic curve. Recently, Stolbunov [27] proposed a Diffie-Hellman type system based on the difficulty of computing isogenies between ordinary elliptic curves, with the stated aim of obtaining quantum-resistant cryptographic protocols. The fastest known (classical) probabilistic algorithm for solving this problem is the algorithm of Galbraith and Stolbunov [10], based on the algorithm of Galbraith, Hess, and Smart [9]. This algorithm is exponential, with a worst-case running time of $O(\sqrt[4]{q})$. However, on a quantum computer, recent work of Childs et al. [5] has shown that the private keys in Stolbunov's system can be recovered in subexponential time. Moreover, even if we only consider classical attacks in assessing security levels, Stolbunov's scheme requires 229 seconds (even with precomputation) to perform a single key exchange operation at the 128-bit security level on a desktop PC [27, Table 1].

In this work we present isogeny-based cryptosystems that address both the performance and security drawbacks of Stolbunov's system. Our scheme achieves performance on the order of one second (cf. Section 6) at the 128-bit security level (as measured against the fastest known quantum attacks) using desktop PCs, making it far faster than Stolbunov's scheme. In terms of security, our scheme is not vulnerable to the algorithm of Childs et al. [5], nor to any algorithm of this type, since it is not based on a group action. The fastest known attacks against our scheme, even on quantum computers, require fully exponential time. Our scheme involves a new computational assumption upon which its quantum resistance is based, and like all new computational assumptions, further study and the passage of time is needed for validation.

Nevertheless, we believe our proposal represents a promising candidate for quantum-resistant isogeny-based public-key cryptography.

Our proposal uses isogenies between *supersingular* elliptic curves rather than ordinary elliptic curves. The main technical difficulty is that, in the supersingular case, the endomorphism ring is noncommutative, whereas Diffie-Hellman type protocols require commutativity. We show how to overcome this obstacle by providing the outputs of the isogeny on certain points as auxiliary input to the protocol. To the best of our knowledge, nothing similar to this idea has ever previously appeared in the literature. Providing this auxiliary input does not seem to make the problem of finding isogenies any easier; see Section 5.2 for a full discussion. The multiple orders of magnitude of performance gains in our scheme arise from the fact that supersingular isogeny graphs are much faster to navigate than ordinary graphs. In Section 5.1 we provide formal statements of the hardness assumptions and security reductions for our system. Finally, in Section 6 we present implementation results confirming the correctness and performance of our protocol.

2 Isogenies

Let E_1 and E_2 be elliptic curves defined over a finite field $\mathbb{F}_q$. An isogeny $\phi : E_1 \to E_2$ defined over $\mathbb{F}_q$ is a non-constant rational map defined over $\mathbb{F}_q$ which is also a group homomorphism from $E_1(\mathbb{F}_q)$ to $E_2(\mathbb{F}_q)$ [23, III.4]. The degree of an isogeny is its degree as a rational map. For separable isogenies, to have degree ℓ means to have kernel of size ℓ. Every isogeny of degree greater than 1 can be factored into a composition of isogenies of prime degree over $\bar{\mathbb{F}}_q$ [6].

An endomorphism of an elliptic curve E defined over $\mathbb{F}_q$ is an isogeny $E \to E$ defined over $\mathbb{F}_{q^m}$ for some m. The set of endomorphisms of E together with the zero map forms a ring under the operations of pointwise addition and composition; this ring is called the endomorphism ring of E and denoted $\mathrm{End}(E)$. The ring $\mathrm{End}(E)$ is isomorphic either to an order in a quaternion algebra or to an order in an imaginary quadratic field [23, V.3.1]; in the first case we say E is supersingular and in the second case we say E is ordinary.

Two elliptic curves E_1 and E_2 defined over $\mathbb{F}_q$ are said to be isogenous over $\mathbb{F}_q$ if there exists an isogeny $\phi\colon E_1 \to E_2$ defined over $\mathbb{F}_q$. A theorem of Tate states that two curves E_1 and E_2 are isogenous over $\mathbb{F}_q$ if and only if $\#E_1(\mathbb{F}_q) = \#E_2(\mathbb{F}_q)$ [29, §3]. Since every isogeny has a dual isogeny [23, III.6.1], the property of being isogenous over $\mathbb{F}_q$ is an equivalence relation on the finite set of $\bar{\mathbb{F}}_q$-isomorphism classes of elliptic curves defined over $\mathbb{F}_q$. Accordingly, we define an isogeny class to be an equivalence class of elliptic curves, taken up to $\bar{\mathbb{F}}_q$-isomorphism, under this equivalence relation.

The ℓ-torsion group of E, denoted $E[\ell]$, is the set of all points $P \in E(\bar{\mathbb{F}}_q)$ such that ℓP is the identity. For ℓ such that $p \nmid \ell$, we have $E[\ell] \cong \mathbb{Z}/\ell\mathbb{Z} \oplus \mathbb{Z}/\ell\mathbb{Z}$.

Curves in the same isogeny class are either all supersingular or all ordinary. Traditionally, most elliptic curve cryptography uses ordinary curves; however, for this work we will be interested in supersingular curves. We assume for the remainder of this paper that we are in the supersingular case.

Supersingular curves are all defined over $\mathbb{F}_{p^2}$, and for every prime $\ell \nmid p$, there exist $\ell+1$ isogenies (counting multiplicities) of degree ℓ originating from any given such supersingular curve. Given an elliptic curve E and a finite subgroup Φ of E, there is up to isomorphism a unique isogeny $E \to E'$ having kernel Φ [23, III.4.12]. Hence we can identify an isogeny by specifying its kernel, and conversely given a kernel subgroup the corresponding isogeny can be computed using Vélu's formulas [32]. Typically, this correspondence is of little use, since the kernel, or any representation thereof, is usually as unwieldy as the isogeny itself. However, in the special case of kernels generated by $\mathbb{F}_{p^2}$-rational points of smooth order, specifying a generator of the kernel allows for compact representation and efficient computation of the corresponding isogeny, as we demonstrate below.

2.1 Ramanujan graphs

Let $G = (\mathcal{V}, \mathcal{E})$ be a finite graph on h vertices $\mathcal{V}$ with undirected edges $\mathcal{E}$. Suppose G is a regular graph of degree k, i.e., exactly k edges meet at each vertex. Given a labeling of the vertices $\mathcal{V} = \{v_1, \ldots, v_h\}$, the

adjacency matrix of G is the symmetric $h \times h$ matrix A whose ij-th entry $A_{i,j} = 1$ if an edge exists between v_i and v_j and 0 otherwise.

It is convenient to identify functions on $\mathcal{V}$ with vectors in $\mathbb{R}^h$ via this labeling, and therefore also think of A as a self-adjoint operator on $L^2(\mathcal{V})$. All of the eigenvalues of A satisfy the bound $|\lambda| \leq k$. Constant vectors are eigenfunctions of A with eigenvalue k, which for obvious reasons is called the trivial eigenvalue λ_{triv}. A family of such graphs G with $h \to \infty$ is said to be a sequence of *expander graphs* if all other eigenvalues of their adjacency matrices are bounded away from $\lambda_{\text{triv}} = k$ by a fixed amount.[3] In particular, no other eigenvalue is equal to k; this implies the graph is connected. A Ramanujan graph is a special type of expander which has $|\lambda| \leq 2\sqrt{k-1}$ for any nontrivial eigenvalue which is not equal to $-k$ (this last possibility happens if and only if the graph is bipartite). The Ramanujan property was first defined in [15]. It characterizes the optimal separation between the two largest eigenvalues of the graph adjacency matrix, and implies the expansion property.

A fundamental use of expanders is to prove the rapid mixing of the random walk on $\mathcal{V}$ along the edges $\mathcal{E}$. The following rapid mixing result is standard but we present it below for completeness. For the proof, see [11] or [7, 14, 22].

Proposition 2.1 *Let G be a regular graph of degree k on h vertices. Suppose that the eigenvalue λ of any nonconstant eigenvector satisfies the bound $|\lambda| \leq c$ for some $c < k$. Let S be any subset of the vertices of G, and x be any vertex in G. Then a random walk of length at least $\frac{\log 2h/|S|^{1/2}}{\log k/c}$ starting from x will land in S with probability at least $\frac{|S|}{2h} = \frac{|S|}{2|G|}$.*

2.2 Isogeny graphs

An isogeny graph is a graph whose nodes consist of all elliptic curves in $\mathbb{F}_q$ belonging to a fixed isogeny class, up to $\overline{\mathbb{F}}_q$-isomorphism (so that two elliptic curves which are isomorphic over $\overline{\mathbb{F}}_q$ represent the same node in the graph). In practice, the nodes are represented using j-invariants, which are invariant up to isomorphism. Isogeny graphs for supersingular elliptic curves were first considered by Mestre [16], and were shown by Pizer [19, 20] to have the Ramanujan property.

Every supersingular elliptic curve in characteristic p is defined over either $\mathbb{F}_p$ or $\mathbb{F}_{p^2}$ [23], so it suffices to fix $\mathbb{F}_q = \mathbb{F}_{p^2}$ as the field of definition for this discussion. Thus, in contrast to ordinary curves, there are a finite number of isomorphism classes of curves in any given isogeny class; this number is in fact the genus g of the modular curve $X_0(p)$, which is roughly $\frac{p+1}{12}$. It turns out that all supersingular curves defined over $\mathbb{F}_{p^2}$ belong to the same isogeny class [16]. For a fixed prime value of $\ell \neq p$, we define the vertices of the supersingular isogeny graph $\mathcal{G}$ to consist of these g isomorphism classes of curves, with edges given by isomorphism classes of degree-ℓ isogenies, defined as follows: two isogenies $\phi_1, \phi_2 \colon E_i \to E_j$ are isomorphic if there exists an automorphism $\alpha \in \text{Aut}(E_j)$ (i.e., an invertible endomorphism) such that $\phi_2 = \alpha\phi_1$. Pizer [19, 20] has shown that $\mathcal{G}$ is a connected $k = \ell + 1$-regular multigraph satisfying the Ramanujan bound of $|\lambda| \leq 2\sqrt{\ell} = 2\sqrt{k-1}$ for the nontrivial eigenvalues of its adjacency matrix.

3 Public-key cryptosystems based on supersingular curves

In this section we present a key-exchange protocol and a public-key cryptosystem analogous to those of [21, 27], using supersingular elliptic curves. Since the discrete logarithm problem is unimportant when elliptic curves are used in an isogeny-based system, we propose using supersingular curves of smooth order to improve performance. In the supersingular setting, it is easy to construct curves of smooth order, and using a smooth order curve will give a large number of isogenies that are fast to compute. Specifically, we fix $\mathbb{F}_q = \mathbb{F}_{p^2}$ as the field of definition, where p is a prime of the form $\ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$. Here ℓ_A and ℓ_B are small primes, and f is a

[3] Expansion is usually phrased in terms of the number of neighbors of subsets of G, but the spectral condition here is equivalent for k-regular graphs and also more useful for our purposes.

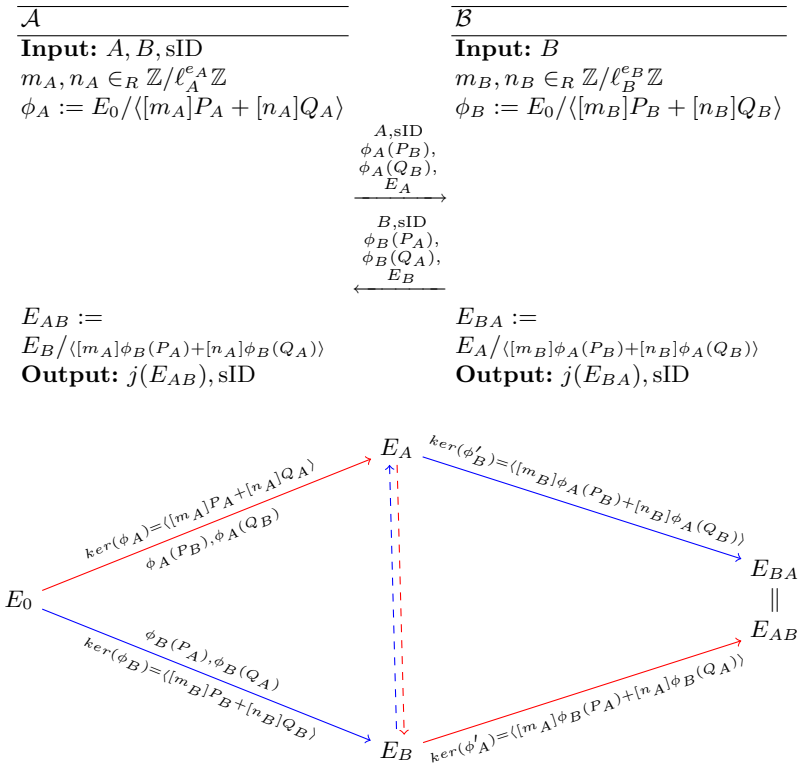

Fig. 1. Key-exchange protocol using isogenies on supersingular curves.

cofactor such that p is prime. Alice and Bob will each take a random walk on a different isogeny graph; Alice will use the graph consisting of isogenies of degrees ℓ_A, and Bob will use the graph of degree ℓ_B isogenies. The main technical modification is that, since ideal classes no longer commute (or indeed even multiply together) in the supersingular case, extra information must be communicated as part of the protocol in order to ensure that both parties arrive at the same common value. This is in contrast to the ordinary case [27], where the existence of an abelian class group allows for the straightforward creation of a Diffie-Hellman type system.

3.1 Key exchange

We fix as public parameters a supersingular curve E_0 defined over $\mathbb{F}_{p^2}$, and bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ which generate $E_0[\ell_A^{e_A}]$ and $E_0[\ell_B^{e_B}]$ respectively, so that $\langle P_A, Q_A\rangle = E_0[\ell_A^{e_A}]$ and $\langle P_B, Q_B\rangle = E_0[\ell_B^{e_B}]$. Alice chooses two random elements $m_A, n_A \in_R \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A, and computes an isogeny $\phi_A \colon E_0 \to E_A$ with kernel $K_A := \langle [m_A]P_A + [n_A]Q_A\rangle$. Alice also computes the image $\{\phi_A(P_B), \phi_A(Q_B)\} \subset E_A$ of the basis $\{P_B, Q_B\}$ for $E_0[\ell_B^{e_B}]$ under her secret isogeny ϕ_A, and sends these points to Bob together with E_A. Similarly, Bob selects random elements $m_B, n_B \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$ and computes an isogeny $\phi_B \colon E_0 \to E_B$ having kernel $K_B := \langle [m_B]P_B + [n_B]Q_B\rangle$, along with the points $\{\phi_B(P_A), \phi_B(Q_A)\}$. Upon receipt of E_B and $\phi_B(P_A), \phi_B(Q_A) \in E_B$ from Bob, Alice computes an isogeny $\phi'_A \colon E_B \to E_{AB}$ having kernel equal to $\langle [m_A]\phi_B(P_A) + [n_A]\phi_B(Q_A)\rangle$; Bob proceeds *mutatis mutandis*. Alice and Bob can then use the common j-invariant of

$$E_{AB} = \phi'_B(\phi_A(E_0)) = \phi'_A(\phi_B(E_0)) = E_0/\langle [m_A]P_A+[n_A]Q_A, [m_B]P_B+[n_B]Q_B\rangle$$

to form a secret shared key. For specific details of how each of the above computations can be performed efficiently, we refer the reader to Section 4.

The full protocol is given in Figure 1. We denote by A and B the identifiers of Alice and Bob, and use sID to denote the unique session identifier.

3.2 Public-key encryption

The key-exchange protocol of Section 3.1 can easily be adapted to yield a public-key cryptosystem, in much the same way that Elgamal encryption follows from Diffie-Hellman. We briefly give the details here. All notation is the same as above. Stolbunov [27] uses a similar construction, upon which ours is based.

Setup: Choose $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$, E_0, $\{P_A, Q_A\}$, $\{P_B, Q_B\}$ as above. Let $\mathcal{H} = \{H_k : k \in K\}$ be a hash function family indexed by a finite set K, where each H_k is a function from $\mathbb{F}_{p^2}$ to the message space $\{0,1\}^w$.

Key generation. Choose two random elements $m_A, n_A \in_R \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$, not both divisible by ℓ_A. Compute $E_A, \phi_A(P_B), \phi_A(Q_B)$ as above, and choose a random element $k \in_R K$. The public key is the tuple $(E_A, \phi_A(P_B), \phi_A(Q_B), k)$ and the private key is (m_A, n_A, k).

Encryption. Given a public key $(E_A, \phi_A(P_B), \phi_A(Q_B), k)$ and a message $m \in \{0,1\}^w$, choose two random elements $m_B, n_B \in_R \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, not both divisible by ℓ_B, and compute

$$h = H_k(j(E_{AB})),$$
$$c = h \oplus m.$$

The ciphertext is $(E_B, \phi_B(P_A), \phi_B(Q_A), c)$.

Decryption. Given a ciphertext $(E_B, \phi_B(P_A), \phi_B(Q_A), c)$ and a private key (m_A, n_A, k), compute the j-invariant $j(E_{AB})$ and set

$$h = H_k(j(E_{AB})),$$
$$m = h \oplus c.$$

The plaintext is m.

4 Algorithmic aspects

We now give specific algorithms to implement the abovementioned steps efficiently.

4.1 Parameter generation

For any fixed choice of $\ell_A^{e_A}$ and $\ell_B^{e_B}$, one can easily test random values of f (of any desired cryptographic size) until a value is found for which $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f - 1$ or $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f + 1$ is prime; the prime number theorem in arithmetic progressions (specifically, the effective version of Lagarias and Odlyzko [13]) provides a sufficient lower bound for the density of such primes.

Once the prime $p = \ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$ is known, Bröker [2] has shown that it is easy to find a supersingular curve E over $\mathbb{F}_{p^2}$ having cardinality $(p \mp 1)^2 = (\ell_A^{e_A}\ell_B^{e_B} \cdot f)^2$. Starting from E, one can select a random supersingular curve E_0 over $\mathbb{F}_{p^2}$ by means of random walks on the isogeny graph (cf. Proposition 2.1); alternatively, one can simply take $E_0 = E$. In either case, E_0 has group structure $(\mathbb{Z}/(p \mp 1)\mathbb{Z})^2$. To find a basis for $E_0[\ell_A^{e_A}]$, choose a random point $P \in_R E_0(\mathbb{F}_{p^2})$ and multiply it by $(\ell_B^{e_B} \cdot f)^2$ to obtain a point P' of order dividing $\ell_A^{e_A}$. With high probability, P' will have order exactly $\ell_A^{e_A}$; one can of course check this by multiplying P' by powers of ℓ_A. If the check succeeds, then set $P_A = P'$; otherwise try again with another P. A second point Q_A of order $\ell_A^{e_A}$ can be obtained in the same way. To check whether Q_A is independent of P_A, simply compute the Weil pairing $e(P_A, Q_A)$ in $E[\ell_A^{e_A}]$ and check that the result has order $\ell_A^{e_A}$; as before, this happens with high probability, and if not, just choose another point Q_A. Note that the choice of basis has no effect on the security of the scheme, since one can convert from one basis to another using extended discrete logarithms, which are easy to compute in $E_0[\ell_A^{e_A}]$ by [30].

Multiplication based

Input: E_0, R_0
1: **for** $0 \leq i < e_A$ **do**
2: $P_i \leftarrow \ell_A^{e_a - i - 1} R_0$;
3: Compute $\phi_i : E_i \to E_i/\langle P_i \rangle$;
4: $E_{i+1} \leftarrow E_i/\langle P_i \rangle$;
5: $R_{i+1} \leftarrow \phi_i(R_i)$;
6: **end for**
Output: E_{e_A}

Isogeny based

Input: E_0, R_0
1: $Q_0 \leftarrow R_0$;
2: **for** $0 \leq j < e_A - 1$ **do**
3: $Q_{j+1} \leftarrow \ell_A Q_j$;
4: **end for**
5: **for** $0 \leq i < e_A$ **do**
6: Compute $\phi_i : E_i \to E_i/\langle Q_{e_A - 1} \rangle$;
7: $E_{i+1} \leftarrow E_i/\langle Q_{e_A - 1} \rangle$;
8: **for** $i \leq j < e_A - 1$ **do**
9: $Q_{j+1} \leftarrow \phi_i(Q_j)$;
10: **end for**
11: **end for**
Output: E_{e_A}

Fig. 2. Key exchange algorithms.

4.2 Key exchange

It remains to describe how Alice and Bob can compute isogenies of a given kernel. We show how to compute $\phi_A \colon E_0 \to E_A$ where $E_A = E_0/\langle [m_A]P_A + [n_A]Q_A \rangle$; the same procedure suffices to compute all the other isogenies mentioned. The computation is performed using a version of Hensel lifting modulo ℓ_A. Let $R_0 := [m_A]P_A + [n_A]Q_A$. The order of R_0 is $\ell_A^{e_A}$. For $0 \leq i < e_A$, let

$$E_{i+1} = E_i/\langle \ell_A^{e_A - i - 1} R_i \rangle, \qquad \phi_i : E_i \to E_{i+1}, \qquad R_{i+1} = \phi_i(R_i),$$

where ϕ_i is a degree ℓ_A isogeny from E_i to E_{i+1}. Then $E_A = E_{e_A}$ and $\phi_A = \phi_{e_A - 1} \circ \cdots \circ \phi_0$.

Figure 2 gives two algorithms for this task. They both compute iteratively $(R_i, \ell_A^{e_A - i - 1} R_i, \phi_i, E_{i+1})$ for $i < e_A$, but they differ in the strategy. The first one, which we will refer to as *multiplication-oriented*, computes at each iteration $\ell_A^{e_A - i - 1} R_i$ from R_i using point addition (or duplication, or triplication). The second one, which we call *isogeny-oriented*, first forms the list $(\ell_A^j R_0)_{j < e_A}$ using point addition, then at each iteration computes the list $(\ell_A^j R_{i+1})_{j < e_A - i - 1}$ by evaluating $\phi_i(\ell_A^j R_i)$ for each j. Observe that Alice and Bob can use one algorithm or the other independently.

A quick analysis shows that both algorithms require $O(\log^2 p)$ operations in $\mathbb{F}_p$. The major cost in the multiplication-based one is scalar point multiplication; this costs $O(e_A \log_2 \ell_A)$ double-and-adds at each iteration and is repeated $e_A \sim \log_{\ell_A} \sqrt{p}$ times. The major cost in the isogeny-based algorithm is the isogeny evaluation at step 8; each evaluation costs $O(\ell_A)$ operations and there are $\frac{1}{2} e_A (e_A - 1)$ of them. By forming the ratio of these quantities, we obtain $O(\log_2 \ell_A / \ell_A)$, so we see that the multiplication-based algorithm is preferable as ℓ_A grows—but we cannot grow ℓ_A indefinitely, because eventually Step 3 becomes the dominant cost. Our implementation, described in Section 6, supports the isogeny-oriented approach for $\ell_A = 2, 3$ and the multiplication-oriented approach for $\ell_A > 2$.

4.3 Isogenies of Montgomery curves

Independently of which method is chosen, it is important to use pick models for elliptic curves that offer the fastest isogeny evaluation performance. The literature on efficient formulas for evaluating small degree isogenies is much less extensive than for point multiplication. In this section we provide explicit and efficient formulas for evaluating isogenies using curves in Montgomery form.

Each of our curves has group structure $(\mathbb{Z}/(p \mp 1)\mathbb{Z})^2$ and its twist has group structure $(\mathbb{Z}/(p \pm 1)\mathbb{Z})^2$. Hence either the curve or its twist has a point of order 4. Consequently, we can write our curves in Montgomery form as follows:

$$E \ : \ B^2 y^2 = x^3 + Ax^2 + x \tag{1}$$

Montgomery curves have very efficient arithmetic on their Kummer line (i.e. by representing points by the coordinates $(X : Z)$ where $x = X/Z$) [17]. The Kummer line identifies P with $-P$, and thus it is not possible to add two distinct points; however it is still possible to compute any scalar multiple of a point. Also observe that since P and $-P$ generate the same subgroup, isogenies can be defined and evaluated correctly on the Kummer line. The goal of this section is to give explicit and efficient formulas for such isogenies.

Let E be the Montgomery curve defined by Eq. (1). It has a point of order two in point $P_2 = (0, 0)$, and a point of order four in $P_4 = (1, \sqrt{(A+2)/B})$—eventualy defined over a quadratic extension—such that $[2]P_4 = P_2$. Montgomery curves have twists of the form $\tilde{y} = \sqrt{c}y$; these are isomorphisms when c is a square. The change of coordinates $\tilde{x} = x/B, \tilde{y} = y/B$ brings the curve E to the Weierstrass form

$$\tilde{E} \ : \ \tilde{y}^2 = \tilde{x}^3 + \frac{A}{B}\tilde{x}^2 + \frac{1}{B^2}\tilde{x},$$

and the point P_4 to $P_4' = (1/B, \ldots)$. Inversely, given a Weierstrass curve $\tilde{E}$ with equation $\tilde{y}^2 = \tilde{x}^3 + a\tilde{x}^2 + b\tilde{x}$, together with a point $P_4 = (1/\beta, \ldots)$—with its ordinate possibly lying in a quadratic extension—such that $[2]P_4 = (0, 0)$, the change of variables $\tilde{x} = x/\beta, \tilde{y} = y/\beta$ brings $\tilde{E}$ to the Montgomery form $\beta y^2 = x^3 + a\beta x^2 + x$.

Let G be a subgroup of the Montgomery curve E of odd cardinality ℓ and let h be the degree $(\ell - 1)/2$ polynomial vanishing on the abscissas of G. With a twist $y = \tilde{y}/\sqrt{B}$, we can put E in the form $\tilde{y}^2 = \tilde{x}^3 + A\tilde{x}^2 + \tilde{x}$, and this doesn't change the abscissas of G or the polynomial h. Now we can use Vélu's formulas to compute an isogeny having G for kernel: this gives an isogeny ϕ and a curve F such that

$$\begin{aligned} F \ &: \ y^2 = x^3 + a_2x^2 + a_4x + a_6, \\ \phi &: E \to F, \\ (x, y) &\mapsto \left(\frac{g(x)}{h(x)^2}, y\sqrt{B}\left(\frac{g(x)}{h(x)^2}\right)' \right). \end{aligned}$$

Because ℓ is odd, the point $(0, 0)$ of E is sent to a point of order two in F. A closer look at Vélu's formulas (see Eq. (3) below) shows that $\phi(0, 0) = (p_{-1} - p_1, 0)$, where p_1 is the sum of the abscissas of G and p_{-1} is the sum of their inverses. By the change of variables $\tilde{x} = x - p_{-1} + p_1$, we bring F to the form $\tilde{F} : \tilde{y}^2 = \tilde{x}^3 + a\tilde{x}^2 + b\tilde{x}$. Now $\phi(P_4)$ is a point of order four (possibly in a quadratic extension). Its abscissa in $\tilde{F}$ is rational and is given by $1/\beta = g(1)/h(1) - p_{-1} + p_1$, so we apply the change of variables $\tilde{x} = \bar{x}/\beta, \tilde{y} = \bar{x}/\beta$ to obtain a Montgomery curve. Finally, we have to twist back the image curve to obtain a curve isogenous over the base field: the twist $\bar{y} = y\sqrt{B}$ cancels with the first one and leaves us with square-root-free formulas. The image curve is

$$B\beta y^2 = x^3 + a\beta x^2 + x. \tag{2}$$

To efficiently evaluate these isogenies (either on the full curve or on the Kummer line) we use [1, Proposition 4.1], which says:

$$\frac{g}{h} = \ell x + p_1 - 2(3x^2 + 2Ax + 1)\frac{h'}{h} - 4(x^3 + Ax^2 + x)\left(\frac{h'}{h}\right). \tag{3}$$

To evaluate at a point (x_0, y_0), we compute $h(x_0), h'(x_0), h''(x_0), h'''(x_0)$; applying Horner's rule, this costs $\sim 2\ell$ multiplications using affine coordinates, or $\sim 3\ell$ using projective coordinates. Then we inject these values in Eq. (3) and in its derivative to evaluate the isogeny, this only takes a constant number of multiplications (plus one inversion in affine coordinates). Finally, the image of (x_0, y_0) is given by

$$\left(\beta(g(x_0)/h(x_0) - p_{-1} + p_1),\ \beta y_0(g/h)'(x_0)\right).$$

Note that if the y-coordinate is not needed[4], we can avoid computing $h'''(x_0)$, thus saving $\sim \ell/2$ multiplications. Of course, for specific small ℓ, such as $\ell = 3, 5$, it is more convenient to write down the isogeny explicitly in terms of the kernel points and find optimized formulas.

[4] While x coordinates are enough to compute Vélu's isogenies and the image curve, this forces the other party to use y-coordinate-free formulas for point multiplication.

When $\ell = 2$, things are more complex, but in our specific case we can easily deal with it. The isogeny of E vanishing $(0,0)$ is readily seen as being

$$F \ : \ y^2 = x^3 + \frac{A+6}{B}x^2 + 4\frac{A+2}{B^2}x, \tag{4}$$

$$\phi : E \to F,$$
$$(x,y) \mapsto \left(\frac{1}{B}\frac{(x-1)^2}{x}, \frac{1}{B}\left(y - \frac{y}{x^2}\right)\right). \tag{5}$$

If a point P_8 satisfying $[4]P_8 = (0,0)$ is known, then $\sqrt{A+2}$ can be computed from the abscissa of $\phi(P_8)$, and F can be put in Montgomery form as before. The isogeny vanishing on a generic point of order two $P_2 \neq (0,0)$ can be easily computed when a point P_4 satisfying $[2]P_4 = P_2$ is known: change coordinates to bring P_2 in $(0,0)$, then use the abscissa of P_4 to express the resulting curve in Montgomery form (this is the same technique as above, taking $\ell = 1$); notice that this step needs to be done at most once per key exchange . When points of order 8 or 4 are not available, as in the last few steps of our setting, ordinary Weierstrass forms yield formulas that require a few extra multiplications.

We conclude this section with operation counts for the key exchange algorithms. We write I, M, S for the costs of one inversion, multiplication and squaring in $\mathbb{F}_{p^2}$ respectively, and we make the assumption $S \leq M$; we count multiplication by constants as normal multiplications. For simplicity, we only list quadratic terms in e_A.

Multiplication-based. If P is a point on the Kummer line, computing P times an n-bit integer costs $(7M + 4S)\log_2 n$ (see [17]). Thus the cumulative cost of Step 2 is

$$\sum_{i=1}^{e_A-1} (7M + 4S)\log_2 \ell_A^i \sim \frac{1}{2}(7M + 4S)(\log_2 \sqrt{p})^2 \log_{\ell_A} 2.$$

Doubling a point on the Kummer line only costs $3M + 2S$, and thus the cost for $\ell_A = 2$ drops down to $\frac{1}{2}(3M + 2S)(\log_2 \sqrt{p})^2$.

Isogeny-based The only quadratic term in e_A appears at Step 8. Since we do not need the y coordinate of the points involved in this step, we only need the values $h(x_0), h'(x_0), h''(x_0)$ in order to apply Eq. (3). We let $s = (\ell_A - 1)/2$ be the degree of h. In affine coordinates, since h is monic, Horner's rule requires $(3s-6)M$, except when $s = 1, 2$. Then, to compute $\beta(g(x_0)/h(x_0) - p_{-1} + p_1)$ we need $I + 8M + 2S$. For $\ell_A = 3$ the total count drops to $I + 6M + 2S$, and for $\ell_A = 5$ it is $I + 8M + 2S$.

In projective coordinates, we first compute $Z, \ldots, Z^s$ at a cost of $(s-1)M$. Then, if $h = \sum_i h_i X^{s-i} Z^i$, we compute the monomials $h_i Z^i$ using sM. Finally we compute h, h', h'' using three applications of Horner's rule, costing again $(3s-6)M$ when $s \neq 1, 2$. The final computation requires $11M + 3S$. For $\ell_A = 3$ the total count is $10M + 2S$, and for $\ell_A = 5$ it is $14M + 3S$.

The difference between the affine and the projective formulas is $I - 2(s-1)M - S$, so the choice between the two must be done according to the ratio I/M.

Finally for $\ell_A = 2$, assuming a point of order 8 on the domain curve is known (which will always be the case, except in the last two iterations), evaluating the x part of Eq. 4 in projective coordinates and bringing the result back to a Montgomery curve costs $2M + S$.

There are $e_A(e_A - 1)$ isogeny evaluations in the algorithm, so, assuming that N is the cost of doing one evaluation, the total cost is about $\frac{1}{2}e_A^2 N = \frac{1}{2}N(\log_2 \sqrt{p})^2(\log_{\ell_A} 2)^2$. We summarize the main costs of the two algorithms in Table 1.

ℓ_A	2	3	5	11	19
$\log_{\ell_A} 2$	1	0.63	0.43	0.29	0.23
Isogeny	$2M + S$	$4.0M + 0.8S$	$1.7M + 0.5S$	$2.0M + 0.2S$	$2.4M + 0.2S$
Multiplication	$3M + 2S$	$4.4M + 2.5S$	$3.0M + 1.7S$	$2.0M + 1.1S$	$1.6M + 0.9S$

Table 1. Comparative costs for the multiplication and isogeny based algorithms using projective coordinates. The entries must be multiplied by $\frac{1}{2}(\log_2 \sqrt{p})^2$ to obtain the full cost.

5 Security

5.1 Complexity assumptions and security proofs

As before, let p be a prime of the form $\ell_A^{e_A}\ell_B^{e_B} \cdot f \pm 1$, and fix a supersingular curve E_0 over $\mathbb{F}_{p^2}$ together with bases $\{P_A, Q_A\}$ and $\{P_B, Q_B\}$ of $E_0[\ell_A^{e_A}]$ and $E_0[\ell_B^{e_B}]$ respectively. In analogy with the case of isogenies over ordinary elliptic curves, we define the following computational problems, adapted for the supersingular case:

Problem 5.1 (Supersingular Isogeny (SSI) problem). Let $\phi_A \colon E_0 \to E_A$ be an isogeny whose kernel is $\langle [m_A]P_A + [n_A]Q_A \rangle$, where m_A and n_A are chosen at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and not both divisible by ℓ_A. Given E_A and the values $\phi_A(P_B)$, $\phi_A(Q_B)$, find a generator R_A of $\langle [m_A]P_A + [n_A]Q_A \rangle$.

We remark that given a generator $R_A = [m_A]P_A + [n_A]Q_A$, it is easy to solve for (m_A, n_A), since E_0 has smooth order and thus extended discrete logarithms are easy in E_0 [30].

Problem 5.2 (Supersingular Computational Diffie-Hellman (SSCDH) problem). Let $\phi_A \colon E_0 \to E_A$ be an isogeny whose kernel is equal to $\langle [m_A]P_A + [n_A]Q_A \rangle$, and let $\phi_B \colon E_0 \to E_B$ be an isogeny whose kernel is $\langle [m_B]P_B + [n_B]Q_B \rangle$, where m_A, n_A (respectively m_B, n_B) are chosen at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ (respectively $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$) and not both divisible by ℓ_A (respectively ℓ_B). Given the curves E_A, E_B and the points $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, $\phi_B(Q_A)$, find the j-invariant of $E_0/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle$.

Problem 5.3 (Supersingular Decision Diffie-Hellman (SSDDH) problem). Given a tuple sampled with probability 1/2 from one of the following two distributions:

- $(E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_{AB})$, where E_A, E_B, $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, $\phi_B(Q_A)$ are as in the SSCDH problem and
$$E_{AB} \cong E_0/\langle [m_A]P_A + [n_A]Q_A, [m_B]P_B + [n_B]Q_B \rangle,$$
- $(E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E_C)$, where E_A, E_B, $\phi_A(P_B)$, $\phi_A(Q_B)$, $\phi_B(P_A)$, $\phi_B(Q_A)$ are as in the SSCDH problem and
$$E_C \cong E_0/\langle [m'_A]P_A + [n'_A]Q_A, [m'_B]P_B + [n'_B]Q_B \rangle,$$
where m'_A, n'_A (respectively m'_B, n'_B) are chosen at random from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ (respectively $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$) and not both divisible by ℓ_A (respectively ℓ_B),

determine from which distribution the triple is sampled.

We conjecture that these problems are computationally infeasible, in the sense that for any polynomial-time solver algorithm, the advantage of the algorithm is a negligible function of the security parameter $\log p$. The resulting security assumptions are referred to as the SSI, SSCDH, and SSDDH assumptions, respectively. Using the methods of Stolbunov [26], it is a routine exercise to prove that the protocols of Section 3 are secure under SSDDH:

Theorem 5.4. *Under the SSDDH assumption, the key-agreement protocol of Section 3.1 is session-key secure in the authenticated-links adversarial model of Canetti and Krawczyk [3].*

Theorem 5.5. *If the SSDDH assumption holds, and the hash function family $\mathcal{H}$ is entropy-smoothing, then the public-key cryptosystem of Section 3.2 is IND-CPA.*

As an illustration of the proof techniques, we provide a proof of Theorem 5.4 in Appendix A.

Remark 5.6. As in the ordinary case [21, 27], our protocols do not provide authentication. One possible workaround for the time being is to use classical public-key authentication schemes in conjunction with the standard observation [24, §6.2] that the authentication step only needs to be secure against the adversary at the time of the initial connection.

5.2 Hardness of the underlying assumptions

Given an SSI (respectively, SSCDH) solver, it is trivial to solve SSCDH (respectively, SSDDH). There is of course no known reduction in the other direction, and given that the corresponding question of equivalence for discrete logarithms and Diffie-Hellman has not yet been completely resolved in all cases, it is reasonable to assume that the question of equivalence of SSI, SSCDH, and SSDDH is at least hard to resolve. For the purposes of this discussion, we will presume that SSI is equivalent to SSDDH.

In the context of cryptography, the problem of computing an isogeny between isogenous supersingular curves was first considered by Galbraith [8] in 1999. The first published cryptographic primitive based on supersingular isogeny graphs is the hash function proposal of Charles et al. [4], which remains unbroken to date (the cryptanalysis of [18] applies only to the LPS graph-based hash function from [4], and not to the supersingular isogeny graph-based hash functions). The fastest known algorithm for finding isogenies between supersingular curves in general takes $O(\sqrt{p}\log^2 p)$ time [4, §5.3.1]; however our problem is less general because the degree of the isogeny is known in advance and is smooth. In addition, the distribution of isogenous curves obtained from taking kernels of the form $\langle [m_A]P_A + [n_A]Q_A\rangle$ is not quite uniform: a simple calculation against Proposition 2.1 indicates that a sequence of e_A isogenies of degree ℓ_A falls short of the length needed to ensure uniform mixing, regardless of the value of p. Since we are the first to propose using isogenies of this type, there is no existing literature addressing the security of the isogenies of the special form that we propose.

There is an easy exponential attack against our cryptosystem that improves upon exhaustive search. To find an isogeny of degree $\ell_A^{e_A}$ between E and E_A, an attacker builds two trees of all curves isogenous to E (respectively, E_A) via isogenies of degree $\ell_A^{e_A/2}$. Once the trees are built, the attacker tries to find a curve lying in both trees. Since the degree of the isogeny ϕ_A is $\sim\sqrt{p}$ (much shorter than the size of the isogeny graph), it is unlikely that there will be more than one isogeny path—and thus more than one match—from E to E_A. Given two functions $f : A \to C$ and $g : B \to C$ with domain of equal size, finding a pair (a, b) such that $f(a) = g(b)$ is known as the *claw problem* in complexity theory. The claw problem can obviously be solved in $O(|A| + |B|)$ time and $O(|A|)$ space on a classical computer by building a hash table holding $f(a)$ for any $a \in A$ and looking for hits for $g(b)$ where $b \in B$. This gives a $O(\ell_A^{e_A/2}) = O(\sqrt[4]{p})$ classical attack against our cryptosystem. With a quantum computer, one can do better using the algorithm in [28], which has complexity $O(\sqrt[3]{|A||B|})$, thus giving an $O(\ell_A^{e_A/3}) = O(\sqrt[6]{p})$ quantum attack against our cryptosystem. These complexities are optimal for a black-box claw attack [33].

We consider the question of whether the auxiliary data points $\phi_A(P_B)$ and $\phi_A(Q_B)$ might assist an adversary in determining ϕ_A. Since (P_B, Q_B) forms a basis for $E_0[\ell_B^{e_B}]$, the values $\phi_A(P_B)$ and $\phi_A(Q_B)$ allow the adversary to compute ϕ_A on all of $E_0[\ell_B^{e_B}]$. This is because any element of $E_0[\ell_B^{e_B}]$ is a (known) linear combination of P_B and Q_B (known since extended discrete logarithms are easy [30]). However, there does not appear to be any way to use this capability to determine ϕ_A. Even on a quantum computer, where finding abelian hidden subgroups is easy, there is no hidden subgroup to find, since ϕ_A has degree $\ell_A^{e_A}$, and thus does not annihilate any point in $E_0[\ell_B^{e_B}]$ other than the identity. Of course, if one could evaluate ϕ_A on arbitrary points of $E_0[\ell_A^{e_A}]$, then a quantum computer could easily break the scheme, and indeed in this case the scheme is also easily broken classically by using a few calls to the oracle to compute a generator of the kernel of the dual isogeny $\hat{\phi}_A$. However, it does not seem possible to translate the values of ϕ_A on $E_0[\ell_B^{e_B}]$ into values on $E_0[\ell_A^{e_A}]$.

	Alice		Bob	
	round 1	round 2	round 1	round 2
$2^{253}3^{161}7-1$	365 ms	363 ms	318 ms	314 ms
$5^{110}7^{91}284-1$	419 ms	374 ms	369 ms	326 ms
$11^{74}13^{69}384-1$	332 ms	283 ms	321 ms	272 ms
$17^{62}19^{60}210+1$	330 ms	274 ms	331 ms	276 ms
$23^{56}29^{52}286+1$	339 ms	274 ms	347 ms	277 ms
$31^{51}41^{47}564-1$	355 ms	279 ms	381 ms	294 ms
$2^{384}3^{242}8-1$	1160 ms	1160 ms	986 ms	973 ms
$5^{165}7^{137}2968-1$	1050 ms	972 ms	916 ms	843 ms
$11^{111}13^{104}78+1$	790 ms	710 ms	771 ms	688 ms
$17^{94}19^{90}116-1$	761 ms	673 ms	750 ms	661 ms
$23^{85}29^{79}132-1$	755 ms	652 ms	758 ms	647 ms
$31^{77}41^{72}166+1$	772 ms	643 ms	824 ms	682 ms
$2^{512}3^{323}799-1$	2570 ms	2550 ms	2170 ms	2150 ms
$5^{220}7^{182}538+1$	2270 ms	2140 ms	1930 ms	1810 ms
$11^{148}13^{138}942+1$	1650 ms	1520 ms	1570 ms	1440 ms
$17^{125}19^{120}712-1$	1550 ms	1430 ms	1520 ms	1380 ms
$23^{113}29^{105}1004-1$	1480 ms	1330 ms	1470 ms	1300 ms

Table 2. Benchmarks for various group sizes and structures.

Finally, we discuss the possibility of adapting the quantum algorithm of Childs et al. [5] for the ordinary case to the supersingular case. For both ordinary and supersingular curves, there is a natural bijection between isogenies (up to isomorphism) and (left) ideal classes in the endomorphism ring. The algorithm of Childs et al. depends crucially on the fact that the ideal classes in the ordinary case form an abelian group. In the supersingular case, the endomorphism ring is a maximal order in a noncommutative quaternion algebra, and the left ideal classes do not form a group at all (multiplication is not well defined). Thus we believe that no reasonable variant of this strategy would apply to supersingular curves.

6 Implementation results and example

We implemented our cryptosystem in the computer algebra system Sage [25] using a mixed C/Cython/Sage architecture. This allows us to access the large palette of number theoretic algorithms distributed with Sage, while still permitting very efficient code in C/Cython for the critical parts such as the algorithms of Section 4.2. The source code can be downloaded from the second author's web page.

Arithmetic in $\mathbb{F}_{p^2}$ is written in C. We use the library GMP for arithmetic modulo p. The field $\mathbb{F}_{p^2}$ is implemented as $\mathbb{F}_{p^2}[X]/(X^2+1)$ (this requires $p = 3 \bmod 4$); using this representation, one multiplication in $\mathbb{F}_{p^2}$ requires three multiplications ($3M$) in $\mathbb{F}_p$, one $\mathbb{F}_{p^2}$ squaring requires two multiplications ($2M$) in $\mathbb{F}_p$, and one $\mathbb{F}_{p^2}$ inversion requires one inversion, two squarings, and two multiplications ($I + 2S + 2M$) in $\mathbb{F}_p$. Our experiments show that, for the sizes we are interested in, $I = 10M$ and $S = 0.8M$.

We implemented the isogeny-based key exchange algorithm for $\ell = 2, 3$ and the multiplication-based algorithm for $\ell > 2$. The main loop is implemented in Cython, while the isogeny evaluation and the Montgomery ladder formulas are written in C.

Finally, the parameter generation is implemented in plain Sage. Because Sage is a collection of many open source mathematical systems, various of its subsystems are involved in this last part. Of these, Pari [31] plays an important role because it is used to compute Hilbert class polynomials and to factor polynomials over finite fields.

All tests ran on a 2.4 GHz Opteron running in 64-bit mode. The results are summarized in Table 2. At the quantum 128-bit security level (768-bit p), our numbers improve upon Stolbunov's reported performance figures [27, Table 1] by over two orders of magnitude (.758 seconds vs. 229 seconds). This is the highest

security level appearing in [27, Table 1], so comparisons at higher levels are difficult. Nevertheless, it seems safe to assume that the improvement is even greater at the 256-bit security level. Our results demonstrate that the proposed scheme is practical.

6.1 Example

As a convenience, we provide an example calculation of a key-exchange transaction. Let $\ell_A = 2$, $\ell_B = 3$, $e_A = 63$, $e_B = 41$, and $f = 11$. We use the starting curve $E_0 : y^2 = x^3 + x$. For the torsion bases, we use

$$\begin{aligned}
P_A &= (237409306833625077410793642140789388589 7i + 252464670185239634930842532821820356969 3,\\
&\quad 194486926041457420622915324351010478172 5i + 130909941321176707805523276846048341720 1)\\
P_B &= (155671603365753087672852505928443176120 6i + 174740732959516524133513164792986606521 5,\\
&\quad 345695620285202883552941999547591538848 3i + 197591287424745857265472071715575500556 6)
\end{aligned}$$

and $Q_A = \psi(P_A), Q_B = \psi(P_B)$, where $i = \sqrt{-1}$ in $\mathbb{F}_{p^2}$ and $\psi(x, y) = (-x, iy)$ is a distortion map [12]. The secret values are

$$\begin{aligned}
m_A &= 2575042839726612324,\ n_A = 8801426132580632841,\\
m_B &= 4558164392438856871,\ n_B = 20473135767366569910
\end{aligned}$$

The isogeny $\phi_A \colon E_0 \to E_A$ is specified by its kernel, and thus the curve E_A is only well defined up to isomorphism; its exact value may vary depending on the implementation. In our case, the curve is $E_A : y^2 = x^3 + ax + b$ where

$$\begin{aligned}
a &= 428128245356224894562061821180718114127i + 214770800990771179013498662460467452576 9\\
b &= 323035926720219746030433183513704240530 93i + 157726433648237019704536235910489488486 2
\end{aligned}$$

and the values of $\phi_A(P_B)$ and $\phi_A(Q_B)$ are

$$\begin{aligned}
\phi_A(P_B) &= (121624303795507829290097485944106602697 6i + 166629113680473868483263718767433090557 2,\\
&\quad 313292160945399836185337294189350010792 3i + 282316493857354948561980003461685523 66)\\
\phi_A(Q_B) &= (203972869442093051915573296501829191066 0i + 242209261432298811249293161552815572738 8,\\
&\quad 168811581269435514554988923851045703427 2i + 137918598460824063891294889034973846753 6)
\end{aligned}$$

Similarly, in our implementation $E_B : y^2 = x^3 + ax + b$ is the curve with

$$\begin{aligned}
a &= 257472239809402296857831386188460894312 2i + 464507557149559062184174132571647427722\\
b &= 286347890751308879214499831122977288619 7i + 176707803671410940579677706508986838675 3
\end{aligned}$$

and the values of $\phi_B(P_A)$ and $\phi_B(Q_A)$ are

$$\begin{aligned}
\phi_B(P_A) &= (251908600334797321477049915416254009818 1i + 145970297400960919872398112545754844087 2,\\
&\quad 207205706793329259932692876625515508138 0i + 891622100638258849401618552145232311395)\\
\phi_B(Q_A) &= (537939945228033932439214329827985436666i + 369874160978813868558848956834319050484 4,\\
&\quad 285386807397180839864966365216121532375 0i + 186973048005362414137237328279585869113 9)
\end{aligned}$$

The common j-invariant of $E_{AB} \cong E_{BA}$, computed by both Alice and Bob, is equal to

$$j(E_{AB}) = 143714549436265511916848280870211141374 4i + 833498096778386452951722285310592056351.$$

7 Conclusion

We propose a new family of conjecturally quantum-resistant cryptographic protocols for key exchange and public-key cryptosystems using isogenies between supersingular elliptic curves of smooth order. In order to compensate for the noncommutative endomorphism rings that arise in this setting, we introduce the idea of providing the images of torsion bases as part of the protocol. Against the fastest known attacks, the resulting scheme improves upon all previous isogeny-based schemes by orders of magnitude in performance at conventional security levels, making it the first practical isogeny-based public-key cryptosystem. Unlike prior such schemes, our proposal admits no known subexponential-time attacks even in the quantum setting.

Acknowledgements

We thank Andrew M. Childs, Alfred Menezes, Vladimir Soukharev, and the anonymous reviewers for helpful comments and suggestions. This work is supported in part by NSERC CRD Grant CRDPJ 405857-10.

References

1. Alin Bostan, François Morain, Bruno Salvy, and Éric Schost. Fast algorithms for computing isogenies between elliptic curves. *Math. Comp.*, 77(263):1755–1778, 2008.
2. Reinier Bröker. Constructing supersingular elliptic curves. *J. Comb. Number Theory*, 1(3):269–273, 2009.
3. Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 453–474. Springer, 2001.
4. Denis Xavier Charles, Kristin E. Lauter, and Eyal Z. Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22:93–113, 2009.
5. Andrew Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time, 2010. `http://arxiv.org/abs/1012.4019/`.
6. Jean-Marc Couveignes. Hard homogeneous spaces, 2006. `http://eprint.iacr.org/2006/291/`.
7. Giuliana Davidoff, Peter Sarnak, and Alain Valette. *Elementary number theory, group theory, and Ramanujan graphs*, volume 55 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2003.
8. Steven D. Galbraith. Constructing isogenies between elliptic curves over finite fields. *LMS J. Comput. Math.*, 2:118–138 (electronic), 1999.
9. Steven D. Galbraith, Florian Hess, and Nigel P. Smart. Extending the GHS Weil descent attack. In *Advances in cryptology—EUROCRYPT 2002 (Amsterdam)*, volume 2332 of *Lecture Notes in Comput. Sci.*, pages 29–44. Springer, Berlin, 2002.
10. Steven D. Galbraith and Anton Stolbunov. Improved algorithm for the isogeny problem for ordinary elliptic curves, 2011. `http://arxiv.org/abs/1105.6331/`.
11. David Jao, Stephen D. Miller, and Ramarathnam Venkatesan. Expander graphs based on GRH with an application to elliptic curve cryptography. *J. Number Theory*, 129(6):1491–1504, 2009.
12. Antoine Joux. The Weil and Tate pairings as building blocks for public key cryptosystems. In *Algorithmic number theory (Sydney, 2002)*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 20–32. Springer, Berlin, 2002.
13. Jeffrey C. Lagarias and Andrew M. Odlyzko. Effective versions of the Chebotarev density theorem. In *Algebraic number fields: L-functions and Galois properties (Proc. Sympos., Univ. Durham, Durham, 1975)*, pages 409–464. Academic Press, London, 1977.
14. Alexander Lubotzky. *Discrete groups, expanding graphs and invariant measures*, volume 125 of *Progress in Mathematics*. Birkhäuser Verlag, Basel, 1994. With an appendix by Jonathan D. Rogawski.
15. Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.
16. Jean-François Mestre. La méthode des graphes. Exemples et applications. In *Proceedings of the international conference on class numbers and fundamental units of algebraic number fields (Katata, 1986)*, pages 217–242, Nagoya, 1986. Nagoya Univ.
17. Peter L. Montgomery. Speeding the pollard and elliptic curve methods of factorization. *Mathematics of Computation*, 48(177):243–264, 1987.
18. Christophe Petit, Kristin Lauter, and Jean-Jacques Quisquater. Full cryptanalysis of LPS and Morgenstern hash functions. In *Proceedings of the 6th international conference on Security and Cryptography for Networks*, SCN '08, pages 263–277, Berlin, Heidelberg, 2008. Springer-Verlag.
19. Arnold K. Pizer. Ramanujan graphs and Hecke operators. *Bull. Amer. Math. Soc. (N.S.)*, 23(1):127–137, 1990.
20. Arnold K. Pizer. Ramanujan graphs. In *Computational perspectives on number theory (Chicago, IL, 1995)*, volume 7 of *AMS/IP Stud. Adv. Math.*, pages 159–178. Amer. Math. Soc., Providence, RI, 1998.
21. Alexander Rostovtsev and Anton Stolbunov. Public-key cryptosystem based on isogenies, 2006. `http://eprint.iacr.org/2006/145/`.
22. Peter Sarnak. *Some applications of modular forms*, volume 99 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 1990.
23. Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1992. Corrected reprint of the 1986 original.

24. Douglas Stebila, Michele Mosca, and Norbert Lütkenhaus. The case for quantum key distribution. In Alexander Sergienki, Saverio Pascazio, and Paolo Villoresi, editors, *Quantum Communication and Quantum Networking: First International Conference, QuantumComm 2009*, volume 36 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. Springer, 2010.
25. William A. Stein et al. *Sage Mathematics Software (Version 4.6.2)*. The Sage Development Team, 2011. `http://www.sagemath.org`.
26. Anton Stolbunov. Reductionist security arguments for public-key cryptographic schemes based on group action. In Stig F. Mjølsnes, editor, *Norsk informasjonssikkerhetskonferanse (NISK)*, pages 97–109, 2009.
27. Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Adv. Math. Commun.*, 4(2):215–235, 2010.
28. Seiichiro Tani. Claw Finding Algorithms Using Quantum Walk. arXiv:0708.2584, March 2008.
29. John Tate. Endomorphisms of abelian varieties over finite fields. *Invent. Math.*, 2:134–144, 1966.
30. Edlyn Teske. The pohlig-hellman method generalized for group structure computation. *Journal of Symbolic Computation*, 27(6):521–534, 1999.
31. The PARI Group, Bordeaux. *PARI/GP, version* `2.4.3`, 2008. available from `http://pari.math.u-bordeaux.fr/`.
32. Jacques Vélu. Isogénies entre courbes elliptiques. *C. R. Acad. Sci. Paris Sér. A-B*, 273:A238–A241, 1971.
33. Shengyu Zhang. Promised and Distributed Quantum Search Computing and Combinatorics. In Lusheng Wang, editor, *Proceedings of the Eleventh Annual International Conference on Computing and Combinatorics*, volume 3595 of *Lecture Notes in Computer Science*, pages 430–439, Berlin, Heidelberg, 2005. Springer Berlin / Heidelberg.

A Proof of Theorem 5.4

We recall the definition of session-key security in the authenticated-links adversarial model of Canetti and Krawczyk [3]. We consider a finite set of *parties* $P_1, P_2, \ldots, P_n$ modeled by probabilistic Turing machines. The adversary $\mathcal{I}$, also modeled by a probabilistic Turing machine, controls all communication, with the exception that the adversary cannot inject or modify messages (except for messages from corrupted parties or sessions), and any message may be delivered at most once. Parties give outgoing messages to the adversary, who has control over their delivery via the Send query. Parties are activated by Send queries, so the adversary has control over the creation of protocol sessions, which take place within each party. Two sessions s and s' are *matching* if the outgoing messages of one are the incoming messages of the other, and vice versa.

We allow the adversary access to the queries SessionStateReveal, SessionKeyReveal, and Corrupt. The SessionStateReveal($\mathfrak{s}$) query allows the adversary to obtain the contents of the session state, including any secret information. The query is noted and $\mathfrak{s}$ produces no further output. The SessionKeyReveal($\mathfrak{s}$) query enables the adversary to obtain the session key for the specified session $\mathfrak{s}$, so long as $\mathfrak{s}$ holds a session key. The Corrupt($\mathsf{P_i}$) query allows the adversary to take over the party P_i, i.e., the adversary has access to all information in P_i's memory, including long-lived keys and any session-specific information still stored. A corrupted party produces no further output. We say a session $\mathfrak{s}$ with owner P_i is *locally exposed* if the adversary has issued SessionKeyReveal($\mathfrak{s}$), SessionStateReveal($\mathfrak{s}$), or Corrupt($\mathsf{P_i}$) before $\mathfrak{s}$ is expired. We say $\mathfrak{s}$ is *exposed* if $\mathfrak{s}$ or its matching session have been locally exposed, and otherwise we say $\mathfrak{s}$ is *fresh*.

We allow the adversary $\mathcal{I}$ a single Test($\mathfrak{s}$) query, which can be issued at any stage to a completed, fresh, unexpired session $\mathfrak{s}$. A bit b is then picked at random. If $b = 0$, the test oracle reveals the session key, and if $b = 1$, it generates a random value in the key space. $\mathcal{I}$ can then continue to issue queries as desired, with the exception that it cannot expose the test session. At any point, the adversary can try to guess b. Let $\text{GoodGuess}^{\mathcal{I}}(k)$ be the event that $\mathcal{I}$ correctly guesses b, and define

$$\text{Advantage}^{\mathcal{I}}(k) = \max\left\{0, \left|\Pr[\text{GoodGuess}^{\mathcal{I}}(k)] - \frac{1}{2}\right|\right\},$$

where k is a security parameter.

The definition of security is as follows:

Definition A.1. *A key exchange protocol Π in security parameter k is said to be* session-key secure *in the authenticated-links adversarial model of Canetti and Krawczyk if for any polynomial-time adversary $\mathcal{I}$,*

Algorithm 1 SSDDH distinguisher

Input: $E_A, E_B, \phi_A(P_B), \phi_A(Q_B), \phi_B(P_A), \phi_B(Q_A), E$

1: $r \xleftarrow{R} \{1, \ldots, k\}$, where k is an upper bound on the number of sessions activated by $\mathcal{I}$ in any interaction.
2: Invoke $\mathcal{I}$ and simulate the protocol to $\mathcal{I}$, except for the r-th activated protocol session.
3: For the r-th session, let Alice send $A, i, E_A, \phi_A(P_B), \phi_A(Q_B)$ to Bob, and let Bob send $B, i, E_B, \phi_B(P_A), \phi_B(Q_A)$ to Alice, where i is the session identifier.
4: **if** the r-th session is chosen by $\mathcal{I}$ as the test session **then**
5: Provide $\mathcal{I}$ as the answer to the test query.
6: $d \leftarrow \mathcal{I}$'s output.
7: **else**
8: $d \xleftarrow{R} \{0, 1\}$.
9: **end if**

Output: d

1. *If two uncorrupted parties have completed matching sessions, these sessions produce the same key as output;*
2. $\text{Advantage}^{\mathcal{I}}(k)$ *is negligible.*

Theorem A.2. *Under the SSDDH assumption, the key-agreement protocol of Section 3.1 is session-key secure in the authenticated-links adversarial model of Canetti and Krawczyk [3].*

Proof. The proof is based on the proof given by Canetti and Krawczyk [3, §5.1] for two-party Diffie-Hellman over $\mathbb{Z}_q^*$. A similar strategy was used by Stolbunov [26] in the case of ordinary elliptic curves.

It has been shown in Section 3 that two uncorrupted parties in matching sessions output the same session key, and thus the first part of Definition A.1 is satisfied. To show that the second part of the definition is satisfied, assume that there is a polynomial-time adversary $\mathcal{I}$ with a non-negligible advantage ε. We claim that Algorithm 1 forms a polynomial-time distinguisher for SSDDH having non-negligible advantage.

To prove the claim, we must show that Algorithm 1 has non-negligible advantage (it is clear that it runs in polynomial time). We consider separately the cases where the r-th session is (respectively, is not) chosen by $\mathcal{I}$ as the test session. If the r-th session is not the test session, then Algorithm 1 outputs a random bit, and thus its advantage in solving the SSDDH is 0. If the r-th session is the test session, then $\mathcal{I}$ will succeed with advantage ε, since the simulated protocol provided to $\mathcal{I}$ is indistinguishable from the real protocol. Since the latter case occurs with probability $1/k$, the overall advantage of the SSDDH distinguisher is ε/k, which is non-negligible.