A Friendly Introduction to Supersingular Isogeny Diffie-Hellman

David Urbanik

March 10, 2017

1 Preface

The purpose of this article is to provide an introduction to the Supersingular Isogeny Diffie-Hellman protocol for interested readers who do not have a strong background in the theory of elliptic curves. Although some familiarity with ordinary Diffie-Hellman protocols and elliptic curve arithmetic will be helpful, in principle a reader of this article need only be comfortable with mathematics at the level of a first course in group theory. Readers of this article should not expect to come away with a detailed understanding of the underlying mathematics, but rather some intuition for the main ideas and governing philosophy behind the protocol.

2 Ordinary Diffie-Hellman

It is worth beginning with a review of the ordinary Diffie-Hellman protocol. In ordinary Diffie-Hellman, Alice and Bob wish to establish a shared secret s over an open channel which is being eavesdropped by an eavesdropper, Eve. The protocol specifies a cyclic group $G = \langle g \rangle$ which is public knowledge. Alice and Bob each choose private integers $0 \leq a, b < |G|$ respectively, compute g^a and g^b, and exchange the results over the open channel. Once they have received each other's results, they can each compute $(g^b)^a = (g^a)^b$, which they take to be their shared secret s. Diagrammatically, this is depicted in Figure 1.

By itself, this is not a description of a secure cryptographic protocol (even against classical adversaries). Rather, the protocol's security depends on the *model* chosen for the group G. For instance, suppose that we choose G to be the integers $\{0, 1, \cdots, n-1\}$ with addition modulo n. Taking $g = 1$, we see that Alice will send a over the open channel, Bob will send b, and the protocol is trivially broken as Eve can easily compute $s = ab \pmod{n}$ herself. Choosing a different generator g will make no difference, as one can invert g mod n to compute $a = g^{-1}(ag) \pmod{n}$ from the public information ag (and likewise for b).

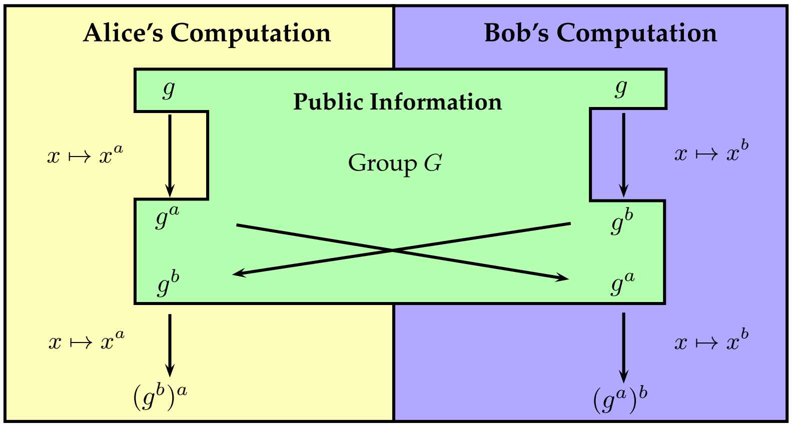

Figure 1: A schematic version of the Diffie-Hellman protocol, emphasizing the public information (in green) and the private information of Alice and Bob (yellow and blue respectively).

Other groups fare better. Suppose we instead choose a prime p, and consider the group $(\mathbb{Z}/p\mathbb{Z})^\times$ of units modulo p. This is again a cyclic group, for which we can find a generator g. Note that this group will in fact be isomorphic to the one in our previous example with $n = p - 1$, yet the best known attacks against Diffie-Hellman over this group are decidedly non-trivial, and achieve only subexponential complexity. For instance, the index calculus methods described in [5] require $L_p(1/3, (128/9)^{1/3})$ expected time to compute the discrete logarithm $a = \log_g(g^a)$ (and hence break the protocol), where

$$L_p(\alpha, c) = \exp((c + o(1))(\log p)^\alpha (\log\log p)^{1-\alpha}).$$

A particularly nice family of groups for Diffie-Hellman is those associated with *elliptic curves* over a field $\mathbb{F}$. In general, elliptic curves are a set of points satisfying an equation cubic in x and quadratic in y, but when the characteristic of $\mathbb{F}$ is not 2 or 3, they can, through various transformations, be converted to a set of the form

$$E(\mathbb{F}) = \{(x, y) \in \mathbb{F}^2 : y^2 = x^3 + ax + b\} \cup \{\infty\},$$

where $a, b \in \mathbb{F}$. Here we have thrown in an extra "point at infinity", denoted ∞, for reasons which we will explain shortly.

Elliptic curves have associated to them a particularly nice group structure, which can be viewed geometrically. This is depicted in Figure 2. If A and B are distinct points on E which do not have the same x-coordinate, the point $A + B$ is obtained by drawing the secant line through A and B, which intersects a 3rd point, and reflecting the result using the map $(x, y) \mapsto (x, -y)$, which is a symmetry of the curve equation. The case for adding A to itself is similar,

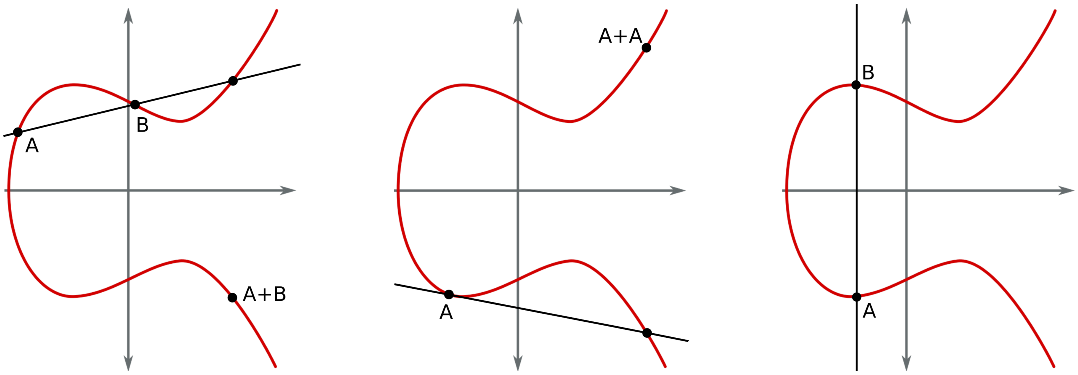

Figure 2: Three images exhibiting the elliptic curve group law on an elliptic curve, drawn in red. In the last image $A + B = \infty$. Although this picture assumes our curve is defined over $\mathbb{R}$, the group law works the same over any field (i.e., the algebraic equations giving the group law are the same), although one can't draw such pictures when viewing the curve over, say, a finite field.

except we need to use a tangent line rather than a secant line. Finally, our point ∞ comes into play when adding points which lie on the same vertical line. Since vertical lines only intersect two ordinary points on elliptic curves, we cannot apply the same rule as in the previous two cases. Our solution is to take these pairs of points to be additive inverses of each other, whose sum $A + B$ is equal to the identity element ∞. One can check that with these rules[1] elliptic curves form an abelian group.

So what about discrete logarithms in elliptic curve groups? The best known attack is based on the *Pollard rho algorithm*, and runs in exponential time $O(\sqrt{|G|})$. It turns out that the Pollard rho algorithm is actually very general – it works in any cyclic group, and only requires that one be able to compute the group operation. Hence, an attack based on Pollard rho can be used against *any* version of the Diffie-Hellman protocol, not just one modelled on elliptic curves. This means that in some sense, elliptic curves appear to be a kind of "ideal model" for Diffie-Hellman. That is, at least conjecturally, one can do no better than elliptic curves for implementing this protocol.

3 Building a new Diffie-Hellman

As you may know, the existence of a universal quantum computer would make it possible to break Diffie-Hellman. Naively, one might suppose that we could simply pick a better model for our group, but the polynomial time discrete logarithm algorithm due to Peter Shor requires nothing more than the ability to

[1] One extra condition that we haven't mentioned is that the curves have to be *non-singular*, which is simply to say that tangent lines have to exist at every point, for this to work.

efficiently compute the group operation, and so all models, even elliptic curves, are affected. That is, Diffie-Hellman is broken by Quantum Computers on *general grounds*, irrespective of the particular implementation chosen.

To understand how we might solve this problem, it is worth revisiting the Diffie-Hellman protocol and asking: what makes it work? For instance, was it important that we used a group G? Not really, since we never needed the existence of inverses or an identity. Was it important that we had a binary operation? Do the maps computed by Alice and Bob have to be exponentiation maps?

What is really needed in a protocol like Diffie-Hellman is actually a kind of *commutative diagram.* That is, one should be able to draw a picture something like the following:

$$\begin{tikzcd}
g \arrow[r, "x\mapsto x^a"] \arrow[d, "x\mapsto x^b"'] & g^a \arrow[d, "x\mapsto x^b"] \\
g^b \arrow[r, "x\mapsto x^a"] & g^{ab}
\end{tikzcd} \tag{1}$$

The diagram is a pictorial representation of a computation that can be done in two different ways. The first way starts with g, follows the top arrow to the right (i.e., computes g^a) and then follows the rightmost arrow downward (computes $(g^a)^b$). The second path works similarly, and computes $(g^b)^a$. To say the diagram *commutes* is simply to say that we end up with the same result either way. This commutativity is important, because it means that Alice and Bob will both get the same result, and hence the same shared secret, from their computations.

The study of mathematics is abound with commutative diagrams. For instance, consider the following suggestively drawn diagram:

$$\begin{tikzcd}
G \arrow[r, "X\mapsto X/A"] \arrow[d, "X\mapsto X/B"'] & G/A \arrow[d, "X\mapsto X/B"] \\
G/B \arrow[r, "X\mapsto X/A"] & G/AB
\end{tikzcd} \tag{2}$$

The diagram depicts, at least on an intuitive level, what happens when we take a group G with two normal subgroups A and B, and "quotient out" A and B in two different orders. The result is, up to isomorphism, the group G/AB (note that $AB = BA$, so the order doesn't matter). If we regard isomorphic groups as being the same, then the diagram commutes.

We can now imagine defining a completely analogous protocol to Diffie-Hellman, which works as follows. The protocol specifies a public group G. Alice and Bob each choose private subgroups A and B respectively. Alice computes the quotient group G/A and sends the result to Bob. Bob computes G/B and sends the result to Alice. They both then compute G/AB (up to isomorphism!) by computing $(G/B)/A$ and $(G/A)/B$ respectively.

This would work, except that the way we have labelled the arrows on this diagram is somewhat misleading. The group A is a subgroup of G, and not a subgroup of G/B, so we cannot quotient out A from G/B. What we really mean by this is to look at the associated quotient map, $\phi_B : G \to G/B$, and to compute $(G/B)/\phi_B(A)$. This distinction is important, because if we wish to use such a diagram to construct a Diffie-Hellman-like protocol, both of the arrows labelled $X \mapsto X/A$ must be computed by Alice with her information, and in general there may be no easy way for Alice to compute $(G/B) \mapsto (G/B)/\phi_B(A)$ if she only knows G/B and A but not the map ϕ_B. We will return to this problem in the next section.

There's one additional thing that needs to be specified for a Diffie-Hellman-like protocol, which is a prescription that certain computations be "easy". For instance, ordinary Diffie-Hellman requires that computing the exponential maps $x \mapsto x^a$ and $x \mapsto x^b$ be easy, since Alice and Bob need to be able to do these computations. In practice, what this really means is that computing the group operation is easy; it's difficult to imagine a group where it's easy to exponentiate but hard to multiply, and if you want to multiply g^k and g^l in $\langle g \rangle$ and *know* that g^k and g^l are the kth and lth powers of g respectively (and you would know this if you had computed these values from g) then computing the product is a matter of computing the exponentiation $g \mapsto g^{k+l}$.

So if ordinary Diffie-Hellman is a combination of the diagram (1) and the prescription that group operations be easy, what is the case for our protocol based on the diagram (2)? Note that although the fact that exponentiation is easy follows if the group operation is easy, it doesn't follow "directly", in the sense that computing $x \mapsto x^a$ by multiplying x by itself a times is still slow. Instead, there is a decomposition of the map $x \mapsto x^a$ in terms of many different applications of, for instance, the maps $y \mapsto y^2$ and $y \mapsto x \cdot y$ (as in the square and multiply algorithm), which "factors" the map $x \mapsto x^a$ into some sequence of maps that can each be computed efficiently. Analogously, we do not require that the computations $X \mapsto X/A$ and $X \mapsto X/B$ in diagram (2) be efficient "directly", but rather that there is a *composition series*, that is a maximal chain of normal subgroups like

$$\{1\} = A_0 \subset A_1 \subset \cdots \subset A_m = A \ , \tag{3}$$

where each successive quotient X/A_1, $(X/A_1)/\phi_{A_1}(A_2)$, etc., can be computed efficiently. Assuming m is reasonably sized, it then follows that the computation $X \mapsto X/A$, which is obtained by applying the m quotients in order, is efficient as well.

In the case of ordinary Diffie-Hellman, we could consider the security of the protocol in the "ideal case", that is, where only the group operation could be presumed efficient[2]. We found that for classical algorithms, the best known attack was the Pollard rho algorithm, which took exponential time, but in the quantum case Shor's algorithm worked in polynomial time. A possible analogue of this "ideal case" for our modified Diffie-Hellman is to assume that "simple

[2] And other auxiliary operations like computations on integers, of course.

quotients" are efficiently computable, that is quotients of the form $X \mapsto X/S$ where S is simple, and to ask what kinds of algorithms, both classical and quantum, can find A from knowledge of X and X/A on this assumption.

There is reason to believe our protocol will be secure against such general attacks. For one, nothing we have said so far has assumed anything in particular about the groups involved, or is even specific to groups! Indeed, diagrams like (2) hold whenever we have a notion of quotient and a version of the first isomorphism theorem (so for instance, rings and modules work too). Therefore, any algorithm which could break such a protocol on purely *general grounds* would have to be extremely powerful – too powerful, one hopes, even for quantum computers.

Of course, just because there is no general way to break a protocol doesn't mean that we can find a feasible, practical implementation. Like with ordinary Diffie-Hellman, it is possible that our choice of model for the group G and its quotients might leave other attack avenues open. For one, we need a model that has exponentially many subgroups A and B of G to choose from, so that the brute force attack of trying all possible quotients is exponential. We also need a way of identifying the quotients up to isomorphism (remember that the diagram (2) only commutes in the sense that the resulting groups are isomorphic), and to resolve the problem of how Alice finds $\phi_B(A)$ from her knowledge of A without knowing ϕ_B, as we discussed earlier.

What's truly remarkable, then, is that it is possible to find a model for this protocol that satisfies all these requirements. What's more, the best known attacks against this particular model, in both the classical and quantum case, are exactly the kind of "ideal attacks" we described – they use nothing more than the efficient computation of the "simple quotient" maps $X \mapsto X/S$. And just as in the case of ordinary Diffie-Hellman, the solution comes from the rich mathematics of elliptic curves.

4 Isogenies and Supersingular Elliptic Curves

We have already discussed elliptic curve groups, so it may come as no surprise that the groups we discussed in the previous section will be modelled on elliptic curves[3]. What remains to be seen is how one can take quotients of these groups, and how we can choose the curves in such a way so that we solve all the potential problems we described in the previous section.

[3] **Technical Remark**: This is somewhat misleading. When we said that diagram (2) commuted what we meant was that it commuted if we identified groups up to isomorphism. We will find here that we have a simple way of identifying elliptic curves up to *elliptic curve* isomorphism, but this is not quite what we were asking for! That's because although isomorphic elliptic curves have isomorphic groups, it is possible for two curves over some field to have isomorphic groups but not be isomorphic as curves (in fact, this will happen all the time in our case). So the diagrams that we will implement here will actually be a sort of refinement of (2), in the sense that there will be more objects available than if we were identifying elliptic curve groups up to *group* isomorphism. Hence we will require the fact that diagrams of the form (2) commute when the isomorphism classes are isomorphism classes of elliptic curves, which is not so obvious.

Note that when we say take a quotient of an elliptic curve group, what we will mean is to construct a quotient group which is *again an elliptic curve.* Hence computing the quotient of a curve E by a subgroup S consists of two parts: one of finding a curve E/S which models the group $E(\mathbb{F})/S$, and another of computing the quotient map $\phi_S : E \to E/S$. The map ϕ_S is what we call an *isogeny.* It turns out that isogenies of elliptic curves are rational functions. The problem of how to compute both the curve E/S and the isogeny map was solved by Velu[6], who gave explicit formulas for both. The general case for an arbitrary isogeny is rather complicated, and takes time polynomial in the size of the quotient group to compute. Since our quotient groups are exponential in size, this justifies our earlier requirement that we can find a chain of subgroups of the form (3), since this will allow us to factor a large isogeny as a composition of many smaller ones, and thus compute the overall quotient efficiently.

Hence, we need to choose our curves in such a way so that both Alice and Bob have a wide range of subgroups to choose from, and so that the subgroups are easily decomposed into chains of the form in (3). The usual solution is to find a large prime of the form $p = 2^{e_A}3^{e_B} - 1$, and take $\mathbb{F} = \mathbb{F}_{p^2}$, a field with p^2 elements[4]. The theory of elliptic curves then tells us that we can find a curve E with group $E(\mathbb{F}_{p^2}) = (\mathbb{Z}/(2^{e_A}3^{e_B})\mathbb{Z})^2$. We then prescribe that Alice's secret keys be cyclic subgroups of order 2^{e_A}, and Bob's secret keys be cyclic subgroups of order 3^{e_B}. If Alice then finds a generator R_A for her subgroup, we then have a composition series of the form in (3):

$$\{\infty\} = \langle 2^{e_A} R_A \rangle \subset \langle 2^{e_A - 1} R_A \rangle \subset \cdots \subset \langle R_A \rangle = A \tag{4}$$

Bob's case is analogous, with 3 replacing 2, e_B replacing e_A, and R_B replacing R_A.

The next thing to explain is the adjective *supersingular.* A supersingular elliptic curve is a very special type of elliptic curve, which is defined by having a particularly large (and non-commutative) *endomorphism ring.* It turns out that computing isogenies between elliptic curves is very closely related to computations in the endomorphism ring of those curves[3]. Hence the larger and more complicated the endomorphism rings of the curves E and E/A are, the more difficult it will be for a potential attacker to discover the isogeny $\phi_A : E \to E/A$ which is Alice's private key (knowledge of the isogeny is equivalent to knowledge of A). For this reason, we choose to use supersingular curves for our protocol; since quotients of supersingular curves remain supersingular, we can do this without issue.

The story so far is summarized in Figure 3. We have two remaining problems to solve. The first is how Alice is able to compute the map $(E/B) \mapsto (E/B)/A$, recalling that what we really meant by this was to compute $(E/B) \mapsto (E/B)/\phi_B(A)$, and the second is how Alice and Bob represent the isomorphism class of the resulting curve so that they end up with the same shared secret.

[4] The general case is $p = f \cdot l_A^{e_A} \cdot l_B^{e_B} \pm 1$, where f is a small factor and $\log(l_A^{e_A}) \approx \log(l_B^{e_B})$, but we will restrict to the most common case here.

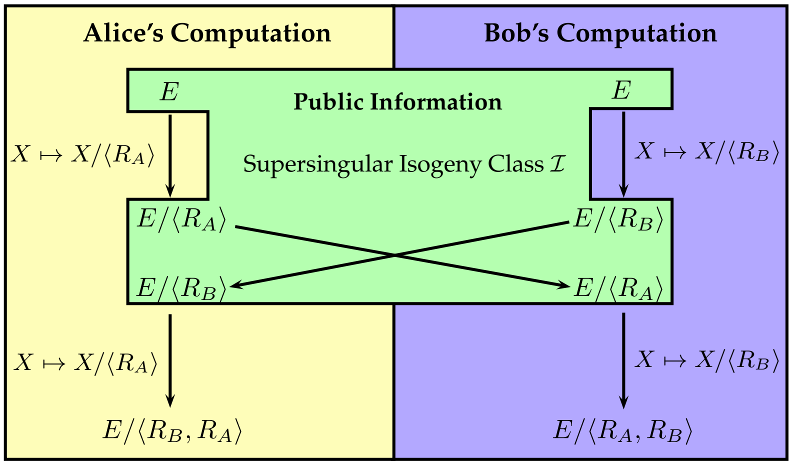
Alice's Computation
Bob's Computation
Public Information
Supersingular Isogeny Class $\mathcal{I}$
E
E
$X \mapsto X/\langle R_A\rangle$
$X \mapsto X/\langle R_B\rangle$
$E/\langle R_A\rangle$
$E/\langle R_B\rangle$
$E/\langle R_B\rangle$
$E/\langle R_A\rangle$
$X \mapsto X/\langle R_A\rangle$
$X \mapsto X/\langle R_B\rangle$
$E/\langle R_B, R_A\rangle$
$E/\langle R_A, R_B\rangle$

Figure 3: What the SIDH protocol looks like "in spirit", drawn in analogy to Figure 1. The analogue of the Group G in Figure 1 is played by the "Supersingular Isogeny Class", which is simply the collection of all supersingular elliptic curves which are related by isogenies.

To solve the first problem, we observe that what Alice really needs is knowledge of the point $\phi_B(R_A) \in \phi_B(A)$, since $\phi_B(A) = \langle \phi_B(R_A)\rangle$ because ϕ_B is a group homomorphism. To solve this, Alice chooses her point R_A as a linear combination $R_A = s_A P_A + t_A Q_A$ of two basis points P_A and Q_A which generate the 2^{e_A}-torsion subgroup of $E(\mathbb{F}_{p^2})$. She keeps the integers s_A and t_A private, and Bob assists her by computing $\phi_B(P_A)$ and $\phi_B(Q_A)$ and sending the result over the public channel. Using this information, Alice can compute $\phi_B(R_A) = s_A\phi_B(P_A)+t_A\phi_B(Q_A)$ (again because ϕ_B is a group homomorphism) and complete the protocol. The case for Bob's computation is analogous. With this modification, our protocol now appears as in Figure 4.

Our final problem is how Alice and Bob compute a common shared secret from the curves $(E/A)/B$ and $(E/B)/A$. It turns out that each elliptic curve has an associated parameter called its *j-invariant*, which determines the curve up to isomorphism. The j-invariant is a rational function of the coefficients of the curve. For instance, if we have a curve $X : y^2 = x^3 + ax + b$, the j-invariant is given by

$$j(X) = 1728\frac{4a^3}{4a^3 + 27b^2}. \tag{5}$$

Since Alice and Bob end up with isomorphic curves, they may take their shared secret s to be $j((E/A)/B) = j((E/B)/A)$, completing the protocol.

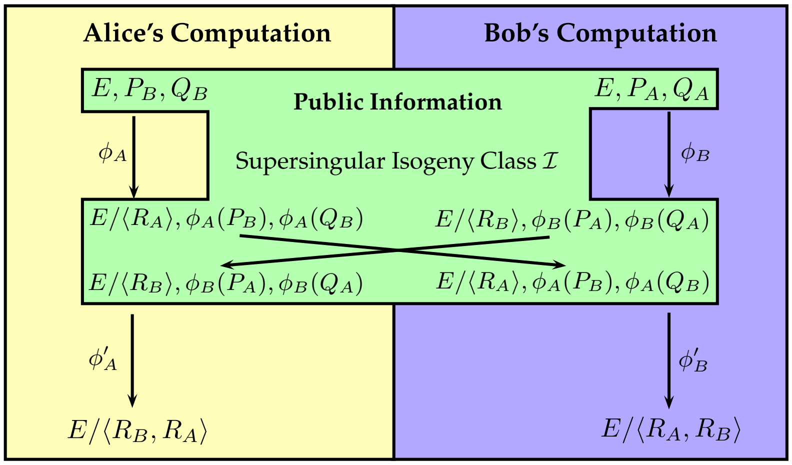
Alice's Computation
Bob's Computation
E, P_B, Q_B
Public Information
E, P_A, Q_A
ϕ_A
Supersingular Isogeny Class $\mathcal{I}$
ϕ_B
$E/\langle R_A\rangle, \phi_A(P_B), \phi_A(Q_B)$
$E/\langle R_B\rangle, \phi_B(P_A), \phi_B(Q_A)$
$E/\langle R_B\rangle, \phi_B(P_A), \phi_B(Q_A)$
$E/\langle R_A\rangle, \phi_A(P_B), \phi_A(Q_B)$
ϕ'_A
ϕ'_B
$E/\langle R_B, R_A\rangle$
$E/\langle R_A, R_B\rangle$

Figure 4: The SIDH protocol in practice. Compare with Figure 1 and Figure 3.

5 Epilogue

There is much more to say about SIDH. The purpose of this article is only to provide intuition, and is not intended as a replacement for a serious reading of the key papers (e.g. [2], [1], [3]). Feel free to email me with any comments, questions or complaints.

References

[1] C. Costello, P. Longa, and M. Naehrig. Efficient algorithms for supersingular isogeny diffie-hellman. In *Lecture Notes in Computer Science*, pages 572–601. Advances in Cryptology CRYPTO 2016, 2016.

[2] L. De Feo, D. Jao, and J. Plut. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Math. Cryptol.*, 2014.

[3] Steven D. Galbraith, C. Petit, B. Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In *Springer LNCS*.

[4] D. Jao. *Handbook of Information and Communication Security*, chapter 3, pages 35–57. Springer, 2010. Avaliable at `http://djao.math.uwaterloo.ca/wiki/images/a/a1/Handbook.pdf`.

[5] A. Joux, R. Lercier, N. Smart, and F. Vercauteren. The number field sieve in the medium prime case. In *Lecture Notes in Computer Science*, pages 326–344. Advances in Cryptology CRYPTO 2006, 2006.

[6] J. Velu. Isogenies entre courbes elliptiques. *CR Acad. Sci. Paris Ser. AB*, 1971. 273:A238A241.