Supersingular Isogeny Diffie-Hellman and related cryptosystems

João Machado

July 25, 2019

Abstract

We present the necessary mathematical background to understand the Supersingular Isogeny Diffie-Hellman and use it to describe the details of the protocol. We present as well an OT framework that is UC secure adapting it from the original so that it works with secret sharing protocols, providing as well its instantiation with the SIDH protocol. Thus we provide an UC secure, OT protocol based on post-quantum cryptography.

Keywords: SIDH, OT-protocol, UC-secure, post-quantum cryptography

1 Introduction

The main object of study for this thesis was the Supersingular Isogeny Diffie-Hellman as proposed in [DFJP14]. One of the protocols to get to the second round of NIST call for submissions of post-quantum cryptography protocols, it is the only one based on elliptic curves. This gives it particular advantages, it is one of the protocols with the smallest key sizes, and disadvantages, it is computationally more expensive than several other protocols. From it we create a OT protocol that is Universally Composable.

We explain to some detail the workings of the Supersingular Isogeny Diffie Hellmann protocol proposed in [DFJP14]. After, we adapt a framework that given an appropriate public key encryption algorithm creates an OT protocol that is UC secure so that we can use it with public key exchange algorithms (and thus SIDH). Finally we create a group action on the space of public keys of SIDH so that we can instantiate the framework with it, thus making the OT protocol UC-secure against quantum treats.

2 Supersingular Elliptic Curves and Isogenies

The SIDH algorithm is based on the difficulty of finding a path between two elliptic curves on the l-isogeny graph (that is finding an l-degree isogeny that maps on onto the other). However, there are several details that needed to come together for it to work. So, before we delve into the protocol lets introduce some notions.

Let $\mathbb{F}_q$ be the finite field with $q = p^2$ elements, $p \neq 2, 3$, and denote by $\overline{\mathbb{F}}_q$ the algebraic closure of $\mathbb{F}_q$. Let as well E/F_q be an elliptic curve,

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

An isogeny is a rational function between elliptic curves that is both a morphism of the curve as a variety and preserves the group structure. As an isogeny is a group morphism we can define both the group of homomorphism between two elliptic curves,

$$\mathrm{Hom}(E_0, E_1) := \{\phi : E_0 \to E_1 \mid \phi \text{ is an isogeny } \}$$

and the endomorphism ring of a curve,

$$\mathrm{End}(E) := \{\phi : E \to E \mid \phi \text{ is an isogeny } \}$$

Most of the literature studies this objects with the isogenies defined over $\overline{\mathbb{F}}_q$ and we will do likewise for the most part. However, there are important results from [Wat69] and [Sch87] where we will want to look for the substructures define over $\mathbb{F}_q$. We will identify those by a subscript $\mathbb{F}$, i.e., $Hom_{\mathbb{F}}(E_0, E_1)$ and $End_{\mathbb{F}}(E)$.

There are two particular examples of isogenies to keep in mind: multiplication by m and the Frobenius morphism ϕ_q. The first, denoted by $[m] : P \mapsto \underbrace{P + \ldots + P}_{m \text{ times}}$ is a completely separable isogeny with degree m^2. In particular we know that $E[m] := \ker[m] = \mathbb{Z}/m\mathbb{Z} \oplus \mathbb{Z}/m\mathbb{Z}$ when $p \not| m$. When $m = p^e$ the structure depends whether the curve is supersingular, having then $E[m] = \{\mathcal{O}\}$, or ordinary, which has $E[m] = \mathbb{Z}/m\mathbb{Z}$.

The frobenius morphism is a completely inseparable isogeny $\phi_q : P \mapsto P^p$, where we denoted $P^p := (X, Y)^p = (X^p, Y^p)$. As it is purely inseparable we have $\ker \phi_q = \{\mathcal{O}\}$. The two important facts about it are that it fixes the rational points of E, i.e., $\ker(\phi_q - 1) = E(\mathbb{F}_q)$, and that its characteristic polynomial depends on the number of such points, i.e., $\phi_q^2 - [t]\phi_q + [q] = 0$ where $t = 1 + q - \#E(\mathbb{F}_q)$ is called the Frobenius trace of E.

Now, we say that E is a *supersingular elliptic curve* if it has any/all of the following equivalent properties:

1. $E[p^r] = \mathcal{O}$ for all $r \geq 1$;
2. ϕ_q é totalmente inseparável;
3. $\mathrm{End}(E)$ is an order in a quaternion algebra over $\mathbb{Q}$, ramified at p and ∞.

Note, in particular, that this means that the Endomorphism ring of a supersingular elliptic curve is non commutative. The next theorem shows that this is not a problem for our protocol:

Proposition 2.1. *Let E_1, E_2, E_3 be elliptic curves over $\mathbb{F}_q$ and suppose there exists separable isogenies $\alpha_2 : E_1 \to E_2$ and $\alpha_3 : E_1 \to E_3$. Then, if $\ker \alpha_2 = \ker \alpha_3$, E_2 is isomorphic to E_3 over $\overline{\mathbb{F}}_q$.*

In particular, this means $j(E_2) = j(E_3)$.

The structure of supersingular elliptic curves is directly connected to the trace of the frobenius morphism:

Theorem 2.2. *Let E be a supersingular elliptic curve defined over a finite field $\mathbb{F}_q$ with $q = p^m$. Let as well t be the trace of the Frobenius endomorphism. The group structure of $E(\mathbb{F}_q)$ is one of the following:*

- *If $t^2 = q, 2q, 3q$, then $E(\mathbb{F}_q)$ is cyclic;*
- *If $t = 0$, then $E(\mathbb{F}_q)$ is either cyclic or isomorphic to $\mathbb{Z}/\frac{q+1}{2}\mathbb{Z} \oplus \mathbb{Z}/2\mathbb{Z}$;*
- *If $t = \mp 2\sqrt{q}$ then $E(\mathbb{F}_q) \simeq \left(\mathbb{Z}/(\sqrt{q} \pm 1)\mathbb{Z}\right)^2$*

Furthermore $t = \mp 2\sqrt{q}$ only if m is even.

We will be interested in the last case $t = \mp 2\sqrt{q}$. Further, if we have in mind that, for curves defined over finite fields, $\#E(\mathbb{F}_q) + \#E_{twist}(\mathbb{F}_q) = 2q + 2$ we see that that case corresponds to curves and their twists according to the signal. This is particularly relevant if we know that

Theorem 2.3. *Two elliptic curves E, E' defined over $\mathbb{F}_q$ are isogenous over $\mathbb{F}_q$ if and only if $\#E(\mathbb{F}_q) = \#E'(\mathbb{F}_q)$.*

Weil Pairing

The Weil e_m-pairing e_m is a function

$$e_m : E[m] \times E[m] \to \mu_m$$

where μ_m denotes the m roots of unity. This function places several constraints in the structure of our elliptic curve. Moreover, it is used for both verifying (and thus attacking) that the public points in the protocols are of the correct form. This comes from the fact that the pairing, apart from the usual properties of bilinearity, being alternating, Galois invariance and non degeneracy, we see that it preserves isogenies:

$$e_m(\phi(S), \phi(T)) = e_m(\hat{\phi}(\phi(S)), T) = e_m([n]S, T) = e_m(S, T)^n \tag{2}$$

3 SIDH

Let us now define the shared set up. We start by assigning to Alice and Bob a small prime each, p_1 and p_2, respectively, and from there generate a prime $p = p_1^{l_1} p_2^{l_2} f \mp 1$, where l_i are of cryptographic size and f is small. This will be the characteristic of our basis field. As we will be using supersingular elliptic curves we take our field $F = \mathbb{F}_{p^2}$ as all supersingular elliptic curves have their j over it. Then, generate a supersingular elliptic curve $E_0/\mathbb{F}_{p^2}$ with $E(\mathbb{F}_{p^2}) = (p \pm 1)^2$, that is, $t^2 = 4q$ (using for example [Brö07]). This guaranties that both $E[l_1^{l_1}]$ and $E[l_2^{l_2}]$ are contained in $E(\mathbb{F}_{p^2})$.Finally we choose two pairs of linearly independent points $P_a, Q_a \in E[p_1^{l_1}]$ and $P_b, Q_b \in E[p_2^{l_2}]$, where by independent we mean that $\langle P_i, Q_i \rangle = E\left[p_i^{l_i}\right]$.

Both Alice and Bob are now in condition to choose their private keys. Alice chooses two integers $0 \leq a_1, a_2 \leq p_1^{l_1}$ such that $p_1 \not| (a_1, a_2)$. Likewise, Bob chooses integers $0 \leq b_1, b_2 \leq p_2^{l_2}$ such that $p_2 \not| (b_1, b_2)$.

From their private keys they generate a point in their torsion subgroups, $G_a = \langle [a_1]P_a + [a_2]Q_a \rangle$ and $G_b = \langle [b_1]P_b + [b_2]Q_b \rangle$. This are the points that are used to define the kernels of their first isogeny. This is the first tricky point as a direct application of Vélu's formulas is very expensive if one remembers that $p_i^{l_i}$ should be of cryptographic size. One must remember Proposition 2.1, which means that we can instead compute l_i isogenies of degree p_i which is less costly. There are some details that we don't mention here, a complete description of this process is given in [DFJP14]. Thus, factorizing the isogeny and using Vélu's formulas they compute the induced separable isogenies ϕ_a, ϕ_b.

Alice and Bob execute now the first message exchange:

$$Alice : (\phi_a(E), \phi_a(P_b), \phi_a(Q_b)) \rightarrow \qquad\qquad \leftarrow (\phi_b(E), \phi_b(P_a), \phi_b(Q_a)) : Bob$$

They can now compute a second isogeny each. Alice, for example, uses $\phi_b(P_a), \phi_b(Q_a)$ to compute $\phi_b([a_1]P_a + [a_2]Q_a) = [a_1]\phi_b(P_a) + [a_2]\phi_b(Q_a)$ thus computing a second isogeny

$$\phi_{ab} : \phi_b(E) \rightarrow \phi_b(E)/\langle \phi_b([a_1]P_a + [a_2]Q_a) \rangle$$

Bob does likewise and computes:

$$\phi_{ba} : \phi_a(E) \rightarrow \phi_a(E)/\langle \phi_a([b_1]P_b + [b_2]Q_b) \rangle$$

The important thing is to notice that, although they need not get the same curve (as Vélu's isogenies are only defined up to isomorphism) we must have $\phi_{ab}(E) \cong \phi_{ba}(E)$ as we see in the following diagram which commutes only up to isomorphism in $\psi(E)$:

$$\begin{tikzcd}
& E/G_a =: E_a \arrow[dr, "\phi_{ba}"] \arrow[dd, dashed, shift right] & \\
E \arrow[ur, "\phi_a"] \arrow[dr, "\phi_b"'] & & E_a/G_b \cong E_b/G_a \\
& E/G_b =: G_b \arrow[ur, "\phi_{ab}"'] \arrow[uu, dashed, shift right] &
\end{tikzcd}$$

This is a direct consequence of Proposition 2.1 as both torsion subgroups have coprime cardinality and so the induced isogenies (which are group homomorphisms) must preserve the other torsion group.

Alice and Bob now share two isomorphic curves, thus, we take their j to be the shared secret. Finally, the fact that $j(E_{ab})$ is sufficiently random comes from the fact that we can see the isogenies as a path of length l_1 over the graph $\mathcal{G}_{p_1}$ followed by a path of length l_2 over the graph $\mathcal{G}_{p_2}$ (or vice-versa), both of which have the same vertex set and both of which are the Ramanujan, i.e., the best possible expanders.

4 An Oblivious Transfer UC-secure Framework

Our first original work is an adaptation of the framework presented in [BDD+17] such that it works with public key sharing protocols and one only needs to provide a group action onto the space of public keys instead of a full group structure on said space.

Setting Up the Framework

From here on let G be a group, KS a key exchange protocol, SE a symmetric encryption protocol and KG a key generating protocol of KS, i.e., it generates (pk, sk) key pairs. Let as well $\star$ denote an action of G on PK, the space of public keys and let $\circ$ denote the process from which one derives the shared secrete out of a pk, sk pair. Finally let $\xrightarrow{\$}$ denote a random sampling.

The functionality we want our framework to realize is:

Definition 4.1 (Functionality $\mathcal{F}_{OT}$)**.** *$\mathcal{F}_{OT}$ interacts with a sender Alice and a receiver Bob. The length of the strings λ is fixed and known to both parties. $\mathcal{F}_{OT}$ proceeds as follows:*

- *Upon receiving a message ($\texttt{sender}, \overrightarrow{x}_0, \overrightarrow{x}_1$) from Alice, where each $\overrightarrow{x}_i \in \{0,1\}^\lambda$, store the tuple $(\overrightarrow{x}_0, \overrightarrow{x}_1)$. Ignore further messages from Alice.*
- *Upon receiving a message ($\texttt{receiver}, c$) from Bob, where $c \in \{0,1\}$, check if a tuple $(\overrightarrow{x}_0, \overrightarrow{x}_1)$ was recorded. If yes, send ($\texttt{received}, \overrightarrow{x}_c$) to Bob, send a delayed output ($\texttt{received}$) to Alice and halt. Otherwise, send nothing to Bob, but continue running.*

For the framework to work We require KS and SE to obey some properties. We will only refer those pertaining to KS as they are the ones we will be interested in showing to be obeyed by $SIDH$. The others can be consulted in the thesis and are fulfilled by using an appropriate SE (such as AES).

Property 4.2. *Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be a two stage PPT adversary interacting with a key exchange scheme KE that computes the following:*

$$\begin{aligned}
&G \to g \\
&\mathcal{A}_1(g) \to (pk_1, pk_2, st) \ : \ g \star pk_1 = pk_2 \\
&KG \to (pk', sk') \\
&pk_i \circ sk' = ss_i \\
&\mathcal{A}_2(pk_1, pk_2, st, pk') = (\widetilde{ss}_1, \widetilde{ss}_2)
\end{aligned}$$

Then

$$Pr[(\widetilde{ss}_1, \widetilde{ss}_2) = (ss_1, ss_2)] \in \text{negl}(K)$$

The second property must also be strengthened so that both the public key generated by the action and the corresponding shared secret are only distinguishable with negligible probability:

Property 4.3. *Let $KG \to (pk, sk)$, $G \to g$ and $pk' = g \star pk$. For every PPT distinguisher $\mathcal{A}_0$ and $\mathcal{A}_1$ it holds that:*

$$|Pr[\mathcal{A}_0(pk) = 1] - Pr[\mathcal{A}_0(pk') = 1]| \in \text{negl}(k)$$

as well as

$$|Pr[\mathcal{A}_1(sk \circ pk) = 1] - Pr[\mathcal{A}_1(sk \circ pk') = 1]| \in \text{negl}(k)$$

Finally we must ask that it is hard to create two pairs of ($\texttt{secret key}, \texttt{public key}$) that generate the same shared secret.

Property 4.4.

$$Pr\left[\begin{array}{c|c} pk \circ sk = pk \circ sk & \begin{array}{c} \mathcal{PK} \to pk \\ \mathcal{A}(ss) \to (r_0, r_1, m_0, m_1) \\ r_0, r_1 \in \mathcal{R} \\ m_0 \neq m_1, m_0, m_1 \in \mathcal{M} \end{array} \end{array}\right] \in \text{negl}(k)$$

The Framework

Thus, once we have a key exchange protocol fulfilling the properties above, the following framework provides an OT protocol which is UC-secure.

Theorem 4.5. *When instantiated with a Public Key Exchange type scheme, paired with a symmetric key encryption scheme Enc fulfilling Properties 4.2,4.3,4.4, the framework presented bellow UC-realizes the ideal functionality $\mathcal{F}_{OT}$.*

The first section of the framework (Part 1) deals with the mechanics of the OT protocol and is almost the same as before. We only needed to change the encryption to be symmetric, with key $\mathsf{sk} \circ \mathsf{pk}$, so as to reflect the changed nature of the instantiating protocols.

OT Framework for Key Sharing Protocols - Part 1

Alice		**Bob**
Input:$m_0, m_1 \in \{0,1\}^\lambda$		Input:$c \in \{0,1\}$
$KG(1^k) \xrightarrow{\$} (pk_a, sk_a)$		$KG(1^k) \xrightarrow{\$} (pk_c, sk_c)$
		$\mathcal{F}_1(sid, s) \rightarrow (q)$
		$pk_{\overline{c}} : pk_0 \star pk_1 = q$
	$\xleftarrow{(sid, s, pk_0)}$	
$\mathcal{F}_1(sid, s) \rightarrow (q)$		
$pk_1 : pk_0 \star pk_1 = q$		
$\{0,1\}^k \xrightarrow{\$} p_0, p_1$		
$\mathcal{F}_2(sid, p_i) \rightarrow (p_i')$		
$m_i' = m_i \oplus p_i'$		
$Enc(sk_a \circ pk_i, p_i) \rightarrow ct_i$		
	$\xrightarrow{(sid, pk_a, m_0', m_1', ct_0, ct_1)}$	
		$Dec(pk_a \circ sk_c, ct_c) \rightarrow p_c$
		$\mathcal{F}_2(sid, p_c) \rightarrow p_c'$
		$m_c = m_c' \oplus p_c'$

As for the second, which is essential for the UC proof, more structural changes needed to be done. Apart from the same change to the encrypting protocol, the most important one is the introduction of a second key pair for Alice. In the original protocol Bob needs to verify that Alice provides a correct encryption of both ω_i. Before that could be achieve just by using the possible public keys. Now, however, he would need to get Alice both of Alice's `shared secrets`. And that would defeat the purpose of the protocol. To solve this we introduce a second key pair for Alice (pk_a', sk_a'), used only on this second round, which is then disclosed to Bob so that he can make the needed verifications without being able to learn both messages.

OT Framework for Key Sharing Protocols

$KG(1^k) \to (pk'_a, sk'_a)$

$\mathcal{F}_3(sid, \omega_i) \to (\omega'_i)$

$\mathcal{F}_4(sid, \omega_0|\omega_1|r_0|r_1) \to ch$

$u_i = \omega'_i \oplus (\omega_{1-i}|sk'_a|r_{1-i})$

$Enc(pk_{bi} \circ sk'_a, \omega_i) \to ct'_i$

$$\xrightarrow{(sid, u_0, u_1, ct'_0, ct'_1)}$$

$Dec(sk_c \circ pk'_a, ct'_c) \to \omega_c$

$\mathcal{F}_3(sid, \omega_c) \to (\omega'_c)$

$(\widetilde{\omega}_{\overline{c}}|\widetilde{sk}'_a|\widetilde{r}_{\overline{c}}) = (u_c \oplus \omega'_c)$

$Enc(pk_{b\overline{c}} \circ \widetilde{sk}'_a, \widetilde{\omega}_{\overline{c}}) \to ct'_{\overline{c}}$

$\mathcal{F}_3(sid, \widetilde{\omega}_{\overline{c}}) \to (\widetilde{\omega}'_{\overline{c}})$

$(\widetilde{\omega}_c|\widetilde{sk'}_a|\widetilde{r}_c) = (u_{\overline{c}} \oplus \widetilde{\omega}_{\overline{c}})$

$Enc(pk_{bc} \circ sk'_a, \widetilde{\omega}_c) \to ct'_c$

$\mathcal{F}_4(sid, \widetilde{\omega}_0|\widetilde{\omega}_1|\widetilde{r}_0|\widetilde{r}_1) \to \widetilde{ch}$

$$\xleftarrow{(sid, \widetilde{ch})}$$

The rest of the protocol is much the same as before and as such so is the UC security proof.

4.1 Instantiating for SIDH

A probabilistic Action

The first method we propose for instantiating the framework constructs a probabilistic action on the space of public keys. By probabilistic we mean that not all elements of the group might create a valid new element on the space of public keys. This means Bob might have to resample the random oracle to get a valid second key which is indistinguishable. However, only after the action is computed does Bob need to send Alice any message thus allowing him to search for a valid one. We will divide action in two parts: the first dealing with transforming the curve, the second transforming the points.

Transforming the curve

Transforming the curve is fairly straightforward. We do need to distinguish the curves more than up to isomorphism, i.e., the j is not enough. But, when restricting ourselves to supersingular elliptic curves (and whose j is neither 0 nor 1728) we have that we only need to check whether we have the right number of racional points or if we should use the twist.

Bob's public key transformation for the curve

$\mathbb{F}_q^\times \xrightarrow{\$} a$

$j' = a^{2c-1} j(E)$

If $j' = 0$ or $j' = 1728$ resample

Construct E' such that $j(E') = j'$

Check if E' is supersingular

If $\#E'(\mathbb{F}_q) = \#E(\mathbb{F}_q)$ set $\psi(E) = E'$

Else set $\psi(E) = E_{twist}$

Transforming the points

The transformations for the points is more convoluted and we were forced to introduce some extra parameters to the key in order to make it work. In simple terms, the process is quite the same as before. Fix the x coordinate of a point, multiply it by the group element and generate the point in the new curve whose x coordinate is the resulting product. This does sound simple. However there are several properties the points need to satisfy. It is to enforce those that we do all the extra steps. In the following explanation we will use $\phi(P), \phi(Q)$ to denote the existing key and $\psi(P), \psi(Q)$ the false key we want to generate. Furthermore we will denote by $\overline{P}$ any point Q such that $[l^l f]Q = P$.

- First, we need to ensure that the new point is of the correct order, i.e., $\psi(P) \in \psi(E)[l_1^{l_1}]$. The only way we found of doing that is using the same method as used in SIDH to generate such points: multiply the point by $[l_1^{l_1} f]$.

check notation

- The previous solution introduces new problems: we need to compute points $\overline{\phi(P)}$ and guaranty that Alice does not distinguish the correct key by having the point already in the correct torsion subgroup. The solution comes in three steps. First we alter the protocol such that the points transferred in the messages need to be multiplied when received, i.e, Bob sends $\overline{\phi}$ which Alice then multiplies by $[l_1^{l_1} f]$ before using. Now, computing a possible $\overline{\phi(P)}$ is easy, $[l_2^{l_2} f^{-1}]P$ (with the inverse taken modulus $l_1^{l_1}$) is a solution. However this point remains in $E[l_1^{l_1}]$ and so Alice can distinguish it from a general $\overline{\phi(P)}$. This is solved by introducing an error $R \in E[l_2^{l_2} f]$ which we sum, that is, we use $\overline{\phi(P)} = [l_2^{l_2} f^{-1}]\phi(P) + R$.

- With the two alterations above we are able to generate $(\psi(P), \psi(Q))$ in $\psi(E)[l_1^{l_1}]$. We need yet to make sure that they generate the full torsion subgroup. For this problem we did not find any easy solution and as such we need to generate a new $\psi(Q)$ if this does happen.

- Finally we only need to make sure that $e() = e()$. This we can indeed force. The discrete log in μ can be computed using Polly-Hellman and, likewise, there is a reduction from the discrete log in $E[l_2^{l_2}]$ to the discrete log in μ. Thus we introduce a new parameter α in the message such that, after computing $\psi(Q)$ using the transformation, Alice does one last multiplication and uses $\psi(P), [\alpha]\psi(Q)$ as the second key.

We can see the protocol in the following table, with the diferences for when Bob chooses either 0 or 1.

```
Point transformation for $c = 0$
$\mathbb{F}_{q^2}^{\times} \xrightarrow{\$} a \ \ \mathbb{F}_2 \xrightarrow{\$} s$
$\phi_B\left(E\right)\left[fp_b^{l_b}\right] \xrightarrow{\$} R$
$r := \left(p_b^{l_b} f\right)^{-1} \mod p_a^{l_a}$
$x = x([r]\phi(P_a) + R) * a$
$\overline{\psi(P_a)} \in \psi(E) : x(\overline{\psi(P_a)}) = x$
$\psi(P_a) = [p_b^{l_b} f]\overline{\psi(P_a)}$
Check if $\psi(P_a)$ has order $p_a^{l_a}$

$\mathbb{F}_{q^2}^{\times} \xrightarrow{\$} b \ \ \mathbb{F}_2 \xrightarrow{\$} s$
$x = x([r]\phi(Q_a) + R) * b$
$\overline{\psi(Q_a)} \in \psi(E) : x(\overline{\psi(Q_a)}) = x$
$\psi(Q_a) = [p_b^{l_b} f]\overline{\psi(Q_a)}$
Check if $\psi(Q_a)$ has order $p_a^{l_a}$
Check if $\langle \psi(P_a), \psi(Q_a) \rangle = \psi(E)[p_a^{l_a}]$

Let $\beta = e_{p_a^{l_a}}(\phi(P_a), \phi(Q_a))$
Let $\gamma = e_{p_a^{l_a}}(\psi(P_a), \psi(Q_a))$
$\alpha := \log_{\gamma}(\beta)$

$\xrightarrow{(\phi(E), [r]\phi(P_a) + R, [r]\phi(Q_a) + R, \alpha)}$
```

```
Point transformation for $c = 1$
$\mathbb{F}_{q^2}^{\times} \xrightarrow{\$} a \ \ \mathbb{F}_2 \xrightarrow{\$} s$
$\phi_B\left(E\right)\left[fp_b^{l_b}\right] \xrightarrow{\$} R$
$r := \left(p_b^{l_b} f\right)^{-1} \mod p_a^{l_a}$
$x = x([r]\phi(P_a) + R) * a^{-1}$
$\overline{\psi(P_a)} \in \psi(E) : x(\overline{\psi(P_a)}) = x$
$\psi(P_a) = [p_b^{l_b} f]\overline{\psi(P_a)}$
Check if $\psi(P_a)$ has order $p_a^{l_a}$

$E(F_{q^2}) \xrightarrow{\$} \overline{\psi(Q_a)}$
$\psi(Q_a) = [p_b^{l_b} f]\overline{\psi(Q_a)}$
Check if $\psi(Q_a)$ has order $p_a^{l_a}$
Check if $\langle \psi(P_a), \psi(Q_a) \rangle = \psi(E)[p_a^{l_a}]$

Let $\beta = e_{p_a^{l_a}}(\phi(P_a), \phi(Q_a))$
Let $\gamma = e_{p_a^{l_a}}(\psi(P_a), \psi(Q_a))$
$\alpha := \log_{\gamma}(\beta)$

$\mathbb{F}_{q^2}^{\times} \xrightarrow{\$} b \ \ \mathbb{F}_2 \xrightarrow{\$} s$
$x = x([\alpha]\psi(\underline{Q_a})') * b$
$\overline{\phi(Q_a)'} \in \phi(E) : x(\overline{\phi(Q_a)'}) = x$
Check if $\phi(Q_a)'$ has order $p_a^{l_a}$
Check if $\langle \phi(P_a), \phi(Q_a)' \rangle = \phi(E)[p_a^{l_a}]$
$\gamma = \log_{\phi(Q_a)} \phi(Q_a)'$

$\xrightarrow{(\psi(E), \overline{\psi(P_a)}, [\alpha]\overline{\psi(Q_a)}, \beta)}$
```

A two steps approach

This second approach is based on a non transitive action, thus creating *a priori* an OT protocol that is only UC secure against semi honest adversaries. However, one can find in [CDSMW09] how to transform an OT protocol secure against semi honest adversaries into one secure against malicious adversaries.

In this case the action is much simpler as we act on the pair of points and in a way such as to force the Weil pairing to be the same. So we want to find $\alpha,\ \beta \in \mathbb{F}_q$ such that, for $m = p_2^{l_2}$:

$$e_m(\alpha P_b, \beta Q_b) = e_m(P_b, Q_b)$$

Now, computing the left hand side gives us:

$$e_m(\alpha P_b, \beta Q_b) = e_m(P_b, Q_b)^{\alpha\beta}$$

And so, for the pairing to agree, we must have $\alpha\beta \equiv_m 1$. Thus, given P, Q we just need to find a random integer $q \in [0, p_1^{l_1}]$ such that $(q, m) = 1$. Then the second public key can be defined as $(\phi(E), q\phi(P), q^{-1}\phi(Q))$ where the inverse is taken modulus m. Notice that the condition $(q, m) = 1$ is almost always true as we will be dealing with $m = p^l$. This gives an action of $\mathbb{Z}_{p'^{l'}}$ on the space of public keys. Instantiating with this action gives us a protocol that is secure against semi-honest adversaries. Using the results from [CDSMW09] we get the UC-security against dishonest adversaries.

References

[BDD$^+$17] Paulo S. L. M. Barreto, Bernardo David, Rafael Dowsley, Kirill Morozov, and Anderson C. A. Nascimento. A framework for efficient adaptively secure composable oblivious transfer in the ROM. *CoRR*, abs/1710.08256, 2017.

[Brö07] Reinier Bröker. Constructing supersingular elliptic curves. 2007.

[CDSMW09] Seung Geol Choi, Dana Dachman-Soled, Tal Malkin, and Hoeteck Wee. Simple, black-box constructions of adaptively secure protocols. In Omer Reingold, editor, *Theory of Cryptography*, pages 387–402, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[DFJP14] Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *J. Math. Cryptol.*, 8(3):209–247, 2014.

[Sch87] René Schoof. Nonsingular plane cubic curves over finite fields. *J. Combin. Theory Ser. A*, 46(2):183–211, 1987.

[Wat69] William C. Waterhouse. Abelian varieties over finite fields. *Ann. Sci. École Norm. Sup. (4)*, 2:521–560, 1969.