Supersingular Isogeny Oblivious Transfer

Paulo Barreto[1], Anderson Nascimento[1], Gláucio Oliveira[2], and Waldyr Benits[3]

[1] University of Washington, Tacoma, USA
pbarreto@uw.edu, andclay@uw.edu
http://directory.tacoma.uw.edu/employee/pbarreto,
http://directory.tacoma.uw.edu/employee/andclay
[2] Institute of Mathematics and Statistics, University of São Paulo, Brazil
glaucioaorj@gmail.com
[3] Naval Systems Analysis Center, Brazilian Navy, Brazil
wbenits@yahoo.com.br

Abstract In this paper we present an *Oblivious Transfer* (OT) protocol that combines an OT scheme together with the *Supersingular Isogeny Diffie-Hellman* (SIDH) primitive. Our proposal is a candidate for *post-quantum* secure OT and demonstrates that SIDH naturally supports OT functionality. We consider the protocol in the simplest configuration of $\binom{2}{1}$-SIOT and analyze the protocol to verify its security.

Keywords: supersingular elliptic curves, isogenies, *supersingular isogeny Diffie-Hellman*, *oblivious transfer*.

1 Introduction

The first notion of Oblivious Transfer (OT) was proposed in [18]. Most, if not all, of cryptography protocols can be based on the notion of OT, under the assumption that an efficient OT scheme is available.

Efficient OT protocols are known in a *quantum*-susceptible scenario [4], where the underlying security assumption is the hardness of computing discrete logarithms or factoring integers.

Additionally, many papers have introduced OT in the context of *quantum* cryptography [3,5,6,16,17,24,29], where the legitimate users manipulate quantum states. The *post-quantum* OT research has gradually increased over time. Thus, some examples could be cited, such as the work done by *Raza Kazmi* [15] and *Vanessa Vitse* [26].

In general, in an $\binom{2}{1}$-OT protocol, a sender sends two messages, say m_a and m_b, and the receiver chooses only one of them (for example, the receiver chooses m_a). At the end of the protocol, the sender does not know which of the messages was chosen, and also the receiver learns nothing about the other message (in this case, m_b).

Our contribuition. According to [12], OT is one of the most important structures in cryptography for the construction of secure protocols. In terms of

application, these types of protocols can be used in electronic auctions processes or even in contract signatures [8] or electronic money transaction schemes [2]. In this work, our main objective was the implementation of $\binom{2}{1}$-SIOT protocol using the SIDH primitives from [10] for the purpose of providing privacy between sender and receiver, at the same time, providing a resistance against the imminent advent of the *quantum* computation.

This paper is organized as follows: Section 2 describes our *Supersingular Isogeny Oblivious Transfer* protocol. In section 3, We discuss about security aspects from $\binom{2}{1}$- SIOT. In section 4, we present a conclusion of the security analysis of the proposed protocol. An implementation of the proposed protocol is presented in section 5. In section 6, we present a performance estimate between some OT protocols. Finally, we conclude this work in section 7. In addition, we use Appendix A to introduce some crucial background about isogenies of elliptic curves. A simplified presentation of the *Supersingular Isogeny Diffie-Hellman* (SIDH) of [10] is shown in appendix B. Furthermore, in Appendix C, we present a brief concept of OT protocol and a simplified form of the protocol present in [4]. In Appendix D, we show some definitions about the process that determines linearly independent points used by our proposed protocol. In Appendix E, there is the possibility of applying a symmetric pairing in the security analysis of the SIOT protocol. At last, in Appendix F we will verify that the proposed protocol is able to share certain points that allow to execute the OT functionality.

2 The $\binom{2}{1}$ - SIOT protocol

In this section, we will see a new scheme called *Supersingular Isogeny Oblivious Transfer* (SIOT) protocol. It is fundamentally inspired on schemes described in [4] and [10]. For readers unfamiliar with issues of isogenies between elliptic curves and OT protocol, we suggest an initial reading in appendices A, B and C.

2.1 Notations

We use the cryptographic primitives of the *Supersingular Isogeny Diffie-Hellman* key exchange protocol (SIDH) from [10]. In this way, the following notations below will be used.

i. $\mathcal{M}$, $\mathcal{K}$, $\mathcal{C}$ $\rightarrow$ Set of all plaintexts, keys and ciphertexts with binary strings of fixed length, respectively;
ii. $p \rightarrow$ A prime such that $p = 3 \bmod 4$;
iii. $\mathbb{F}_{p^2} \rightarrow$ A quadratic extension of $\mathbb{F}_p$, where $\mathbb{F}_{p^2} = \mathbb{F}_p[i]/\langle i^2 + 1\rangle$;
iv. $E_0[\mathbb{F}_{p^2}] \rightarrow$ A supersingular elliptic curve over $\mathbb{F}_{p^2}$;
v. $\mathbb{Z}/\ell\mathbb{Z} \rightarrow$ A field of integers modulo ℓ, where ℓ is prime and $\ell \nmid p$;
vi. $P_A, Q_A \rightarrow$ Linearly independent points over the supersingular elliptic curve $E_0[\ell_A^{e_A}]$;
vii. $P_B, Q_B \rightarrow$ Linearly independent points over the supersingular elliptic curve $E_0[\ell_B^{e_B}]$;

viii. $\phi_A, \phi_B \rightarrow$ Isogenies between E_0 and E_A, E_0 and E_B, respectively;
ix. $\phi_A^{'}, \phi_B^{'} \rightarrow$ Isogenies between E_B and E_{BA}, E_A and E_{AB}, respectively;
x. $G_A, H_A \rightarrow$ Images of P_B and Q_B under Sender's private isogeny ϕ_A;
xi. $G_B, H_B \rightarrow$ Images of P_A and Q_A under Receiver's private isogeny ϕ_B;
xii. $\jmath(E_{AB}), \jmath(E_{BA}) \rightarrow \jmath - invariants$ of supersingular elliptic curve E_{AB} and $E_{BA}, respectively$;
xiii e_A, $e_B \rightarrow$ Positive integers;
xiv. $r_A, r_B \rightarrow$ Integers from $\mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and $\mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, respectively;
xv. $f \rightarrow$ Small cofactor to ensure a prime $p = \ell_A^{e_A}\ell_B^{e_B} f \pm 1$;
xvi. $\mathcal{H} \rightarrow$ Hash function such that $\mathcal{H} = \{H_k : k \in K\}$. It is a hash function family indexed by a finite set K, where each H_k is a function from $\mathbb{F}_{p^2}$.
xvii. $\mathcal{E}$, $\mathcal{D} \rightarrow$ Encryption and Decryption algorithms, respectively;
xviii. pk_A, $\mathsf{pk}_B \rightarrow$ Sender's public key and Receiver's public key, respectively;
xiv. sk_A, $\mathsf{sk}_B \rightarrow$ Sender's private key and Receiver's private key, respectively.
xx. $\mathcal{O} \rightarrow$ Special point located at infinity. It acts as a neutral element in the operation of adding points of an elliptic curve over a finite field;
xxi. $ker(\phi) \rightarrow$ Kernel of an isogeny. In particular, it is a finite subgroup of an elliptic curve over closed field $\mathbb{F}_{p^2}$. It can also be denoted by $\langle P, Q\rangle$, where $P, Q \in E[\bar{\mathbb{F}}_{p^2}]$.

2.2 Public parameters

Let E_0 be a supersingular elliptic curve defined over $\mathbb{F}_{p^2}$. For convenience, assume a prime p of form[4] $p = \ell_A^{e_A}\ell_B^{e_B} - 1$ with $\ell_A = 2$ and $e_A \geqslant 4$ (and $f = 1$), or $p = 4\ell_A^{e_A}\ell_B^{e_B} - 1$, where both ℓ_A and ℓ_B are odd primes (and $f = 4$). Hence, either of these choices yield $p = 3 \pmod 4$, enabling the representation[5] $\mathbb{F}_{p^2} = \mathbb{F}_p[i]/\langle i^2 + 1\rangle$ and ensuring that the curve $E_0[\mathbb{F}_{p^2}] : y^2 = x^3 + x$ is supersingular, with group order $(\ell_A^{e_A}\ell_B^{e_B} f)^2$. Additionally, let P_A, Q_A, P_B, Q_B points. Then, $E_0[\ell_A^{e_A}]$ and $E_0[\ell_B^{e_B}]$ are generated by kernel $\langle P_A, Q_A\rangle$ and $\langle P_B, Q_B\rangle$, respectively. The appendix D presents some definitions used to compute such linearly dependent points.

2.3 Premises

1. Let $(\mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme according to definitions 1 and 2 of [4]. The shared symmetric key is the value of the invariant $\jmath$ between two supersingular and isomorphic[6] elliptic curves. We will see in figure 1, that the invariant $\jmath$ will be submitted to hash function $\mathcal{H} = \{H_k : k \in \mathcal{K}\}$ indexed by a finite set $\mathcal{K}$, where each H_k is a function of $\mathbb{F}_{p^2}$;
2. Alice wants to encrypt two messages m_0, $m_1 \in \mathcal{M}$ and send them to Bob. In turn, Bob will decrypt only one of these two messages and Alice will not be aware of his choice;

[4] See [19] that reports a deep research about the choice of SIDH-Friendly Primes.
[5] See [13] for more details about this type of representation.
[6] See [21], Proposition III.1.4(b).

3. Alice and Bob use a *coin-flipping* protocol from [27] to share a single uniform random string of w bits. This ensures that neither Alice nor Bob can guess in advance or control the value of w. Thus, they must use, for instance, a hash function in this bit string to get the linearly independent points U and V. Otherwise, they must generate a new string w.

2.4 Protocol

Figure 1 shows an abstraction of the proposed protocol operation in its simplest form, i.e, $\binom{2}{1}$ - SIOT.

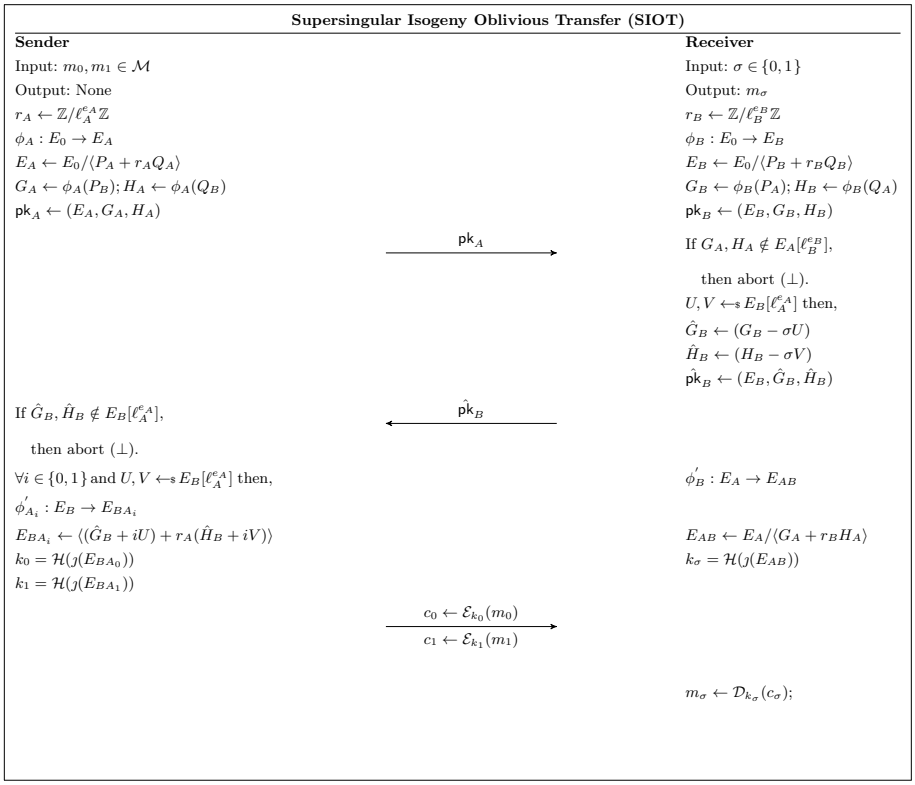


Figure 1. $\binom{2}{1}$ - SIOT protocol.

2.4.1 Generation of key pairs

Setup - Sender

1. Alice secretly chooses a value $r_A \leftarrow Z/\ell_A^{e_A} Z$;
2. She computes:

(a) $ker(\phi_A) = \langle P_A + r_A Q_A \rangle$;
(b) $\phi_A : E_0 \rightarrow E_A$;
(c) $\phi_A(P_B) = G_A$; $\phi_A(Q_B) = H_A$.

3. Alice creates a pair of keys $sk_A = (\phi_A, r_A)$ and $pk_A = (E_A, G_A, H_A)$, i.e, her private and public keys, respectively;
4. Alice sends to Bob $pk_A = (E_A, G_A, H_A)$. He checks if $G_A, H_A \in E_A[\ell_B^{e_B}]$, i.e, $\ell_B^{e_B} G_A = \ell_B^{e_B} H_A = \mathcal{O}_A \in E_A$, since $\{P_B, Q_B\} \subset E_0[\ell_B^{e_B}]$. If this check is valid then, Bob will accept the public key of Alice. Otherwise, that public key will be rejected ($\perp$).

It is denoted $E_A = E_0/\langle P_A + r_A Q_A \rangle$, where $|ker(\phi_A)| = |\langle P_A + r_A Q_A \rangle| = \ell_A^{e_A}$, *i.e*, a separable[7] isogeny of degree $\ell_A^{e_A}$.

Setup - Receiver

1. Bob secretly chooses a value $r_B \leftarrow Z/\ell_B^{e_B} Z$;
2. He computes:
   (a) $ker(\phi_B) = \langle P_B + r_B Q_B \rangle$;
   (b) $\phi_B : E_0 \rightarrow E_B$;
   (c) $\phi_B(P_A) = G_B$; $\phi_B(Q_A) = H_B$;
   (d) For a unique $\sigma \in \{0, 1\}$, $\hat{G}_B = (G_B - \sigma U)$ and $\hat{H}_B = (H_B - \sigma V)$.
3. Bob creates a key pair $sk_B = (\phi_B, r_B)$ and $\hat{pk}_B = (E_B, \hat{G}_B, \hat{H}_B)$, i.e, his private and public key, respectively;
4. He sends $\hat{pk}_B$ to Alice. Then, she performs two checks:
   – If $\hat{G}_B, \hat{H}_B \in E_B[\ell_A^{e_A}]$, i.e, $\ell_A^{e_A} G_B = \ell_A^{e_A} H_B = \mathcal{O}_B \in E_B$ since $\{P_A, Q_A\} \subset E_0[\ell_A^{e_A}]$;
   – If the points U, $V \in E_B[\ell_A^{e_A}]$. This ensure that the pair of points $(G_B + U, H_B + V)$ and $(G_B - U, H_B - V)$ are generated by $E_B[\ell_A^{e_A}]$. Otherwise, $\hat{pk}_B$ will be rejected ($\perp$) and the protocol is restarted to execute another bit string w.

2.4.2 Generation of secret keys

Setup - Sender

1. Alice computes:
   (a) $\forall i \in \{0, 1\}$, $ker(\phi'_{A_i}) = \langle (\hat{G}_B + iU) + r_A(\hat{H}_B + iV) \rangle$;
   (b) $\phi'_{A_i} : E_B \rightarrow E_{BA_i}$;
   (c) $\jmath_i \leftarrow \jmath(E_{BA_i})$;
   (d) $k_i = \mathcal{H}(\jmath_i)$.

Setup - Receiver

1. Bob computes:
   (a) $ker(\phi'_B) = \langle G_A + r_B H_A \rangle$;
   (b) $\phi'_B : E_B \rightarrow E_{AB}$;
   (c) $\jmath_\sigma \leftarrow j(E_{AB})$;
   (d) $k_\sigma = \mathcal{H}(\jmath_\sigma)$.

[7] See [28], Proposition 12.8

2.4.3 Encryption and Decryption

1. $\forall i \in \{0,1\}$, Alice encrypts m_i. Then, $c_i \leftarrow \mathcal{E}(k_i, m_i)$. After that, she sends (c_0, c_1) to Bob;
2. He decrypts and gets $m_\sigma \leftarrow \mathcal{D}(k_\sigma, c_\sigma)$;

3 Security analysis of the $\binom{2}{1}$-SIOT protocol

In the next sections, we will present some definitions for security analysis of the proposed protocol.

3.1 Preliminaries

Definition 1. *A function $\epsilon(\cdot)$ is negligible in $\underline{n}$, or just negligible, if for every positive polynomial $p(\cdot)$ and all sufficiently large $\underline{n}$ it holds that $\epsilon(\cdot) < 1/p(\cdot)$.*

Definition 2. *A probability ensemble $\mathcal{X} = \{\mathcal{X}(n, a)\}$ is an infinite sequence of random variables indexed by $n \in \mathbb{N}$ and $a \in \{0,1\}^*$. The value $\underline{n}$ will represent a security parameter and $\underline{a}$ will represent the parties' inputs.*

Definition 3. *Two distribution ensembles $\mathcal{X} = \{\mathcal{X}(n, a)\}$ and $\mathcal{Y} = \{\mathcal{Y}(n, a)\}$ are said to be computationally indistinguishable, denoted by $\mathcal{X} \overset{\mathbf{c}}{\equiv} \mathcal{Y}$, if for every non-uniform polynomial-time algorithm $\mathcal{D}$ there exists a negligible function $\epsilon(\cdot)$ such that for every $n \in \mathbb{N}$ and $a \in \{0,1\}^*$. Then, $|\Pr[\mathcal{D}(\mathcal{X}(n, a)) = 1] - \Pr[\mathcal{D}(\mathcal{Y}(n, a)) = 1]| \leqslant \epsilon(n)$.*

3.2 Computational problems of isogenies between supersingular elliptic curves

In this section, we will see some cases of computational problems from supersingular elliptic curves that were adapted from [10]. Therefore, let a supersingular curve E_0 over $\mathbb{F}_{p^2}$ together with independent 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. Furthermore, recall that p is a prime of the form defined on section 2.2.

Problem 1 (Decisional Supersingular Isogeny (DSSI) problem). Let $E_A[\mathbb{F}_{p^2}]$ be another supersingular curve. Decide whether E_A is $\ell_A^{e_A}$-isogenous to E_0.

Problem 2 (Computational Supersingular Isogeny (CSSI) problem). Let $\phi_A : E_0 \to E_A$ be an isogeny whose kernel is $R_A = \langle [m_A]P_A + [r_A]Q_A \rangle$ for some $m_A, r_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$. Given the public key (E_A, G_A, H_A). Determine R_A.

Problem 3 (Supersingular Computational Diffie-Helmann (SSCDH) problem). Let $\phi_A : E_0 \to E_A$ be an isogeny whose kernel is $R_A = \langle [m_A]P_A + [r_A]Q_A \rangle$ for some $m_A, r_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and let $\phi_B : E_0 \to E_B$ be an isogeny whose kernel is $R_B = \langle [m_B]P_B + [r_B]Q_B \rangle$ for some $m_B, r_B \in \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$. Given the public keys (E_A, G_A, H_A) and (E_B, G_B, H_B). Determine the $\jmath - invariant$ of $E_0/\langle [m_A]P_A + [r_A]Q_A, [m_B]P_B + [r_B]Q_B \rangle$

Problem 4 (Supersingular Decision Diffie-Hellman (SSDDH) problem). Given a tuple sampled with probability 1/2 from one of the following two distributions:

1. $(E_A, E_B, G_A, H_A, G_B, H_B, E_{AB})$ where E_A, E_B, G_A, H_A, G_B *and* H_B are as in the SSCDH problem (*Problem 3*) then,

$$E_{AB} \simeq E_0/\langle [m_A]P_A + [r_A]Q_A, [m_B]P_B + [r_B]Q_B \rangle,$$

2. $(E_A, E_B, G_A, H_A, G_B, H_B, E_C)$ where E_A, E_B, G_A, H_A, G_B *and* H_B are as in the SSCDH problem (*Problem 3*) then,

$$E_C \simeq E_0/\langle [\hat{m}_A]P_A + [\hat{r}_A]Q_A, [\hat{m}_B]P_B + [\hat{r}_B]Q_B \rangle$$

Let $m_A, r_A, \hat{m}_A, \hat{r}_A \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and $m_B, r_B, \hat{m}_B, \hat{r}_B \in \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$. Determine from which distribution the tuple is sample.

Remark 1. Each sample has a probability 1/2. Thus, for definition 3, we have that:

$$\{E_A, E_B, G_A, H_A, G_B, H_B, E_{AB}\} \overset{\mathrm{c}}{\equiv} \{E_A, E_B, G_A, H_A, G_B, H_B, E_C\}$$

3.3 Notations for security analysis

In the security analysis of the proposed protocol, the followings notations will be used:

i. Application of the $V\acute{e}lu's^8$ $formula \to V\acute{e}lu's$ $formula\{ker(\phi), E\}$;
ii. According to [14] in an OT protocol, the replacement of either m_0 or m_1 by another message m must go unnoticed by the receiver. Let $\tau \in \mathcal{M}$ and $\sigma \in \{0, 1\}$. Then, Alice's view in executing an OT protocol is denoted by $\{\Omega_{Alice}(Alice(1^n, \tau), Bob(1^n, \sigma))\}$, where a security parameter is defined by 1^n. Similarly, we denote Bob's view for $\{\Omega_{Bob}(Alice(1^n, \tau), Bob(1^n, \sigma))\}$;
iii. When Alice or Bob acts like a dishonest user, we denote them by Alice* and Bob*, respectively.

3.4 Some requirements for security analysis

A priori, any secure protocols should resist to any adversarial attack. Thus, to prove that an OT protocol is secure, [12] state that the most important requirements in any security protocol are *correctness* and *privacy*.

[8] See [11], Corollary 25.1.7.

3.4.1 Correctness

Suppose that both Alice and Bob are honest parties taking the $\binom{2}{1}$-SIOT protocol. Let $\sigma, i \in \{0,1\}$ such that $\sigma = i$. Thus, the *correctness* follows the identities below.

$$\begin{aligned}
\jmath(E_{BA_i}) \simeq & \jmath(E_B/\langle (G_B - \sigma \cdot U + i \cdot U) + r_A \cdot (H_B - \sigma \cdot V + i \cdot V)\rangle) \\
\simeq & \jmath(\phi_B(E_0)/\langle (\phi_B(P_A) - \sigma \cdot U + i \cdot U) + r_A(\phi_B(Q_A) - \sigma \cdot V + i \cdot V)\rangle) \\
\simeq & \jmath(\phi_B(E_0)/\langle \phi_B(P_A) + r_A \cdot \phi_B(Q_A)\rangle) \\
\simeq & \jmath(\phi_B(E_0)/\langle \phi_B(P_A + r_A \cdot Q_A)\rangle) \\
\simeq & \jmath(\phi'_{A_i}(\phi_B(E_0))) \simeq \jmath(\phi_A(E_0)/\langle \phi_A(P_B + r_B \cdot Q_B)\rangle) \\
\simeq & \jmath(\phi'_{A_i}(\phi_B(E_0))) \simeq \jmath(E_B/\langle \phi_A(P_B) + r_B \cdot \phi_B(Q_B)\rangle) \\
\simeq & \jmath(\phi'_{A_i}(\phi_B(E_0))) \simeq \jmath(E_B/\langle G_A + r_B \cdot H_A\rangle) \simeq \jmath(\phi'_B(\phi_A(E_0))) \simeq \jmath(E_{AB}).
\end{aligned}$$

3.4.2 Privacy

In the $\binom{2}{1}$-SIOT protocol, Bob's choice should not be known to Alice. Moreover, at the end of the protocol execution, Bob will not be able to gain any knowledge about the message that he did not decrypt. It should be noted that this privacy stems from the difficulty of solving the computational problems seen in section 3.2. Finally, to complement the security proof of the proposed protocol, Theorem 1 was elaborated as follows:

Theorem 1. *Assume that CSSI, SSCDH, SSDDH problems are hard in a group $E(\mathbb{F}_{p^2})$. Then, $\binom{2}{1}$-SIOT protocol ensures privacy between two parties.*

Proof. The proof is to adapt definition 2.6.1 of [12] to $\binom{2}{1}$-SIOT protocol for compatibility with the computational problems mentioned in section 3.2. Let two messages, m_0 and m_1, between two parts (Alice and Bob). An OT protocol is private if the following requirements are valid:

i. ***Non-triviality**: If Alice and Bob follow the protocol correctly then, after an execution in which Alice has for input any m_0, $m_1 \in \mathcal{M}$ and Bob has input bit $\sigma \in \{0,1\}$, the output of Bob is m_σ. In other words, Bob receives pk_A and the pair (c_0, c_1) from Alice. Recalling that $pk_A \leftarrow (E_A, G_A, H_A)$ and $c_\sigma \leftarrow \mathcal{E}(k_\sigma, m_\sigma)$ are well defined. Thus, non-triviality follows from the fact that*

$$\begin{gathered}
V\acute{e}lu's\, formula\{\langle G_A + r_B H_A\rangle, E_A\} \Rightarrow E_{AB} \therefore \\
\mathcal{H}(\jmath(E_{AB})) \Rightarrow k_\sigma.
\end{gathered}$$

Therefore, Bob recovers k_σ implying $m_\sigma \leftarrow \mathcal{D}(k_\sigma, c_\sigma)$ such that σ is a unique binary value secretly chosen by him. Furthermore, upon receiving pk_A, Bob will not be able to compute Alice's private key, i.e, $\mathsf{sk}_A = (\phi_A, r_A)$. If that could be possible, there would be a violation of the CSSI problem difficult hypothesis.

ii. ***Privacy in the case of a dishonest Bob:*** *Let* $\hat{pk}_B \leftarrow (E_B, \hat{G}_B, \hat{H}_B)$ *denotes Bob's public key sent to Alice. Recall that* $\hat{G}_B \leftarrow G_B$, $\hat{H}_B \leftarrow H_B$, *if* $\sigma = 0$ *and* $\hat{G}_B \leftarrow (G_B - U)$, $\hat{H}_B \leftarrow (H_B - V)$, *if* $\sigma = 1$. *In addition, there is Alice's public key* $pk_A := (E_A, G_A, H_A)$ *sent to Bob and a unique value of* $\jmath - invariant$ $\jmath_\sigma = \jmath(Vélu's\, formula\{\langle G_A + r_B H_A\rangle, E_A\})$ *computed by Bob upon receiving Alice's well-defined public key* pk_A. *After that,* $\forall i \in \{0,1\}$, *Alice will compute* $\jmath_i = \jmath(Vélu's\, formula\{\langle(\hat{G}_B + iU) + r_A(\hat{H}_B + iV)\rangle, E_B\})$, *i.e,* $\jmath_0$ *and* $\jmath_1$. *Moreover, Alice will share a unique secret key with Bob. Thus, Alice's privacy is based on the following fact: Bob cannot compute both values of the invariants* $\jmath_0$ *and* $\jmath_1$ $(\jmath_0 \neq \jmath_1)$, *if the hypothesis of the SSCDH problem is difficult. In other words, Bob will be able to compute a unique invariant* $\jmath_\sigma$.

Let $\sigma \in \{0,1\}$ *an auxiliary input and another input with tuple* m_0, m_1, $m \in \mathcal{M}$. *Thus, another way to view the Alice's privacy is that Bob's first message, denoted by* $Bob^*\,(1^n, \sigma)$, *determines whether it should receive* m_0 *or* m_1. *For example, if it determines that it should receive* m_0, *then its view when Alice's input is* (m_0, m_1) *is indistinguishable from its view when Alice's input is* (m_0, m). *Evidently, this implies that Bob cannot learn anything about* m_1 *when it receives* m_0 *and* vice versa. *Hence,*

$$\{\Omega_{Bob^*}(Alice(1^n,(m_0,m_1)); Bob^*(1^n,\sigma))\}_{n\in\mathbb{N}} \overset{\mathbf{c}}{\equiv} \{\Omega_{Bob^*}(Alice(1^n,(m_0,m)); Bob^*(1^n,\sigma))\}_{n\in\mathbb{N}}$$

or

$$\{\Omega_{Bob^*}(Alice(1^n,(m_0,m_1)); Bob^*(1^n,\sigma))\}_{n\in\mathbb{N}} \overset{\mathbf{c}}{\equiv} \{\Omega_{Bob^*}(Alice(1^n,(m,m_1)); Bob^*(1^n,\sigma))\}_{n\in\mathbb{N}}$$

iii. ***Privacy in the case of a dishonest Alice:*** *Note that this requirement shows that Alice cannot distinguish Bob's possible secret choices, i.e, when bit* σ *is set to 0 or 1. In other words, she simply visualizes a* pk_B *public key. Then, the receive's privacy will be checked by following Lemma:*

Lemma 1. *Alice by inputting* $\hat{pk}_B$ *cannot guess* σ *with probability greater than* $1/2 + \epsilon(n)$, *for some negligible function* $\epsilon(n)$ *and* $\forall\ n \in \mathbb{N}$

Proof. We can assume that by receiving $\hat{pk}_B$ *and not knowing the value of Bob's bit* σ, *Alice cannot distinguish the pairs of tuples* $\{(E_B, G_B, H_B)\}$ *and* $\{(E_B, (G_B - U), (H_B - V)\}$, *i.e, for some* $\hat{G}_B, \hat{H}_B \in E_B[\ell_A^{e_A}]$ *such that* $\Pr[(E_B, G_B, H_B) = (E_B, \hat{G}_B, \hat{H}_B)] = \Pr[(E_B, G_B - U, H_B - V) = (E_B, \hat{G}_B, \hat{H}_B))]$ *which is independent of* σ *. Thus, the difficulty of this indistinguishability between these tuples is based on SSDDH problem. We have that:*

$$\{(E_B, G_B, H_B)\} \overset{\mathbf{c}}{\equiv} \{E_B, (G_B - U), (H_B - V)\}$$

According to [20], a distinguisher is a probabilistic algorithm that describes the advantages of an adversary's advantage. Then, suppose that, by contradiction, there is a probabilistic distinguisher Θ *of polynomial time and a non-negligible function* ϵ *such that* $\forall n \in \mathbb{N}$,

$$|P_r[\Theta(E_B, G_B, H_B) = 1] - P_r[\Theta(E_B, (G_B - U), (H_B - V)\}) = 1]| \geqslant \epsilon(n),$$

Then, by subtracting and adding the following term,

$$P_r[\Theta(E_B, (G_B - U - R), (H_B - V - S)) = 1]^9$$

We have that
$|P_r[\Theta(E_B, G_B, H_B) = 1] - P_r[\Theta((E_B, (G_B - U), (H_B - V)) = 1]| \leqslant |P_r[\Theta(E_B, G_B, H_B) = 1] - P_r[\Theta(E_B, (G_B - U - R), (H_B - V - S)) = 1]| + |P_r[\Theta(E_B, (G_B - U - R), (H_B - V - S)) = 1] - P_r[\Theta(E_B, (G_B - U), (H_B - V)) = 1]|$

By contradiction, We suppose that

$$|P_r[\Theta(E_B, G_B, H_B) = 1] - P_r[\Theta(E_B, (G_B - U - R), (H_B - V - S)) = 1]| \geqslant \frac{\epsilon(n)}{2} \quad (3.1)$$

or

$$|P_r[\Theta(E_B, (G_B - U - R), (H_B - V - S)) = 1] - P_r[\Theta(E_B, (G_B - U), (H_B - V)) = 1]| \geqslant \frac{\epsilon(n)}{2} \quad (3.2)$$

Suppose that (3.1) holds. Thus, we can construct a distinguisher $\tilde{\theta}$ for the SSDDH problem that works as follow: Upon input $\hat{pk}_B \leftarrow \{(E_B, (G_B - U), (H_B - V))\}$, the distinguisher $\tilde{\Theta}$ randomly chooses the pair of points R, S. Hence, $\hat{pk}'_B \leftarrow \{(E_B, (G_B - U - R), (H_B - V - S))\}$. On the other hand, if $\hat{pk}_B \leftarrow \{(E_B, G_B, H_B)\}$ then, $\hat{pk}'_B \leftarrow \{E_B, (G_B - R), (H_B - S)\}$. Note that the pairs of points U and V are not used in the last tuple $\hat{pk}'_B$. However, these points as points R and S from to the same group $E_B[\ell_A^{e_A}]$ and could also be randomly chosen by $\tilde{\Theta}$, say points U and V. Thus, we can have that $\hat{pk}'_B \leftarrow \{(E_B, G_B, H_B)\}$ and

$$|P_r[\tilde{\Theta}(E_B, G_B, H_B) = 1] - P_r[\tilde{\Theta}(E_B, (G_B - U), (H_B - V)) = 1]| = |P_r[\Theta(E_B, G_B, H_B) = 1] - P_r[\Theta(E_B, (G_B - U - R), (H_B - V - S)) = 1]| \geqslant \frac{\epsilon(n)}{2},$$

in contradiction to the SSDDH problem. An analogous analysis follows in the case where (3.2) holds. The proof of Bob's privacy is concluded by noting that $\{(E_B, G_B, H_B)\}$, $\{(E_B, (G_B - U), (H_B - V))\}$, regardless of the value of σ, are indistinguishable in Alice's view. In other words, let $\tau \in \{0,1\}^*$ be an auxiliary input. Thus,

$$\{\Omega_{Alice^*}(Alice^*(1^n, \tau), Bob(1^n, 0))\} \stackrel{\mathbf{c}}{\equiv} \{\Omega_{Alice^*}(Alice^*(1^n, \tau), Bob(1^n, 1))\}.$$

According to Lemma 1, the privacy of Bob follows from SSDDH problem over the group $E_B[\ell_A^{e_A}]$. □

Therefore, the above requirements are related to the computacional problems of isogenies in supersingular elliptic curves and they ensure the privacy of the $\binom{2}{1}$-SIOT protocol □

[9] $R, S \in E_B[\ell_A^{e_A}]$.

3.5 Algebraic security analysis of the $\binom{2}{1}$-SIOT protocol

Considering the case of a dishonest Alice, she will use a *Weil* pairing-based distinguisher for trying to find out the secret value σ from honest Bob. In the second situation, the roles will be inverted, *i.e*, Alice will be considered an honest sender and Bob a dishonest receiver. Thus, an analysis is performed in such a way that some algebraic conditions must be obeyed so that Bob is not able to decipher both of Alice's messages.

3.5.1 Preventing a *Weil* pairing-based distinguisher from a possible Alice's dishonesty

Considering the situation where Alice* (the dishonest sender) receiving the information $(E_B, \hat{G}_B, \hat{H}_B)$ from Bob, *a priori*, does not know whether to receive (E_B, G_B, H_B) or $(E_B, G_B - U, H_B - V)$. Alice might consider using the *Weil* pairing to distinguish between these two values.

In what follows, all pairings have order $\ell_A^{e_A}$. After all, the correct points $G_B = \phi_B(P_A)$ and $H_B = \phi_B(Q_A)$ are known to satisfy $e(G_B, H_B) = e(P_A, Q_A)^{\ell_A^{e_A}}$: if this relation does not hold for both of $(\hat{G}_B, \hat{H}_B)$ or $(\hat{G}_B + U, \hat{H}_B + V)$, it would reveal which key Bob has chosen. More generally, because Alice can add any multiple of (U, V) to $(\hat{G}_B, \hat{H}_B)$ and look for such a mismatch, one must have $e(\hat{G}_B + \lambda U, \hat{H}_B + \lambda V) = e(G_B, H_B) = e(P_A, Q_A)^{\ell_A^{e_A}}$ for any $\lambda \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$. Recalling that $U, V, G_B, H_B \in E_B[\ell_A^{e_A}]$ then, they can be wrote as a linear combination, i.e, $U = \alpha G_B + \beta H_B$, $V = \gamma G_B + \delta H_B$ such that $\alpha, \beta, \gamma, \delta \in \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$. Thus, this condition means that:

$$
\begin{aligned}
e(G_B + \lambda U, H_B + \lambda V) &= e(G_B + \lambda\alpha G_B + \lambda\beta H_B, H_B + \lambda\gamma G_B + \lambda\delta H_B) \\
&= e((1+\lambda\alpha)G_B + \lambda\beta H_B, \lambda\gamma G_B + (1+\lambda\delta)H_B) \\
&= e((1+\lambda\alpha)G_B, \lambda\gamma G_B) \\
&\cdot\ e((1+\lambda\alpha)G_B, (1+\lambda\delta)H_B) \\
&\cdot\ e(\lambda\beta H_B, \lambda\gamma G_B) \\
&\cdot\ e(\lambda\beta H_B, (1+\lambda\delta)H_B) \\
&= 1 \\
&\cdot\ e(G_B, H_B)^{(1+\lambda\alpha)(1+\lambda\delta)} \\
&\cdot\ e(H_B, G_B)^{\lambda\beta\lambda\gamma} \\
&\cdot\ 1 \\
&= e(G_B, H_B)^{(1+\lambda\alpha)(1+\lambda\delta)-\lambda^2\beta\gamma} \\
&= e(G_B, H_B),
\end{aligned}
$$

hence it is necessary that $(1+\lambda\alpha)(1+\lambda\delta)-\lambda^2\beta\gamma = 1 \pmod{\ell_A^{e_A}}$, or equivalently $\lambda(\alpha+\delta) + \lambda^2(\alpha\delta - \beta\gamma) = 0 \pmod{\ell_A^{e_A}}$. This must hold for *any* choice of λ, in particular those that are invertible mod $\ell_A^{e_A}$, and hence it must hold that $\lambda(\alpha\delta - \beta\gamma) = -(\alpha+\delta) \pmod{\ell_A^{e_A}}$. Once more, this can only hold for *any* λ

if $\alpha\delta - \beta\gamma = 0 \pmod{\ell_A^{e_A}}$ and $\alpha + \delta = 0 \pmod{\ell_A^{e_A}}$, or equivalently, $\delta = -\alpha \pmod{\ell_A^{e_A}}$ and $\alpha^2 + \beta\gamma = 0 \pmod{\ell_A^{e_A}}$. Therefore, in principle, such conditions should be obeyed to avoid Alice to find out Bob's choice. □

3.5.2 Possible decryptions from a possible dishonest Bob

Recalling $U, V \in E_B[\ell_A^{e_A}]$ are linearly independent points, and write $U = \alpha G_B + \beta H_B$, $V = \gamma G_B + \delta H_B$. Suppose Alice receives an information $(E_B, \hat{G}_B, \hat{H}_B)$ from Bob*. Then, Alice will compute actually the degree-$\ell_A^{e_A}$ isogeny $\phi'_{A_0} : E_B \to E_{BA_0}$ whose kernel is $ker(\phi'_{A_0}) = \langle G_B + r_A H_B \rangle$ and $\phi'_{A_1} : E_B \to E_{AB_1}$ whose kernel is $ker(\phi_{A_1}) = \langle (G_B + U) + r_A(H_B + V) \rangle$. It should be noted[10] that if $ker(\phi'_{A_0}) \subsetneqq ker(\phi_{A_1})$ then, E_{BA_0} is isomorphic to E_{BA_1} *i.e.*, $E_{BA_0} \cong E_{BA_1}$. Moreover, if ϕ_{A_1} is separable then there is a unique isogeny $\hat{\phi_A} : E_{BA_0} \to E_{BA_1}$. Now $(G_B + U) + r_A(H_B + V) = (G_B + \alpha G_B + \beta H_B) + r_A(H_B + \gamma G_B + \delta H_B) = (1 + \alpha + \gamma r_A)G_B + (r_A + \beta + \delta r_A)H_B$. Hence, by inspection, this point can only be in $\langle G_B + r_A H_B \rangle$ with the following conditions:

1. $(1 + \alpha + \gamma r_A)$ is invertible mod $\ell_A^{e_A}$ (i.e. if $\ell_A \nmid 1 + \alpha + \gamma r_A$);
2. $(r_A + \beta + \delta r_A)/(1 + \alpha + \gamma r_A) = r_A \pmod{\ell_A^{e_A}}$, which means $\gamma r_A^2 + (\alpha + \delta) r_A - \beta = 0 \pmod{\ell_A^{e_A}}$ and hence $\gamma r_A^2 + (\alpha - \delta) r_A - \beta = 0 \pmod{\ell_A}$. Thus, a simple constraint on the coefficients ensures that the last equation has no solution then, just force $\ell_A \mid \gamma$ and $\ell_A \mid (\alpha - \delta)$, but $\ell_A \nmid \beta$.

Therefore, it is important that this equation has no solution because, otherwise, if Alice and Bob* cannot control the coefficients α, β, γ, δ apart from ensuring conditions as above, Bob* could be able to decrypt both messages from Alice. □

3.5.3 Wrapping up the conditions

In this section we will consider the three conditions on α, β, γ, and δ based on the equations obtained in sections 3.5.1 and 3.5.2 to ensure $\binom{2}{1}$-SIOT protocol security in a scenario where Alice and Bob are dishonest parties. Thus, conditions on α, β, γ, and δ are obtained that guarantee that Alice will not be able to get the secret choice of Bob's bit b and he will not be able to decipher both pairs c_0 and c_1 sent by Alice. Combining these relations yields $\gamma = -\alpha^2/\beta \pmod{\ell_A^{e_A}}$ since β is certainly invertible mod $\ell_A^{e_A}$. In particular, this means $V = -(\alpha/\beta)U$.

Additionally, in appendix E we will see the application of a symmetric pairing to analyze other possible conditions relative to the coefficients of points U and V. Moreover, appendix F shows the process of sharing of these points.

4 Conclusion of the security of the $\binom{2}{1}$-SIOT protocol

The security proof of the SIOT protocol is based on three parts, namely:

[10] See Theorem 9.6.18 from [11] and Proposition 12.12 from [28].

i. The inherent security characteristics of [10], that is, the computational problems mentioned in section 3.2;

ii. The privacy between a sender and receiver in a communication channel by means of Theorem 1;

iii. An algebraic analysis that used *Weil*'s pairing that defined some conditions necessary for a dishonest sender to does not cheat the security against an honest receive and *vice versa*.

5 Implementation of the $\binom{2}{1}$-SIOT protocol

The $\binom{2}{1}$-SIOT protocol was implement in the *phyton* language, using a MacBook Air with a 1.6 GHz Intel Core i.5 processor, 4GB, 1.600MHz and DDR3 memory. Table 1 shows the values of the prime numbers p that were used in the implementation of the proposed protocol.

Table 1. Values used for p in the $\binom{2}{1}$-SIOT protocol.

$p = \ell_A^{e_A} \ell_B^{e_B} f \pm 1$	Value	Size (bits)
$(3^4 \cdot 5^3 \cdot 4) - 1$	40.499	16
$(3^6 \cdot 5^3 \cdot 4) - 1$	364.499	19
$(3^7 \cdot 5^4 \cdot 4) - 1$	5.467.499	23
$(3^{11} \cdot 5^6 \cdot 4) - 1$	11.071.687.499	34

In this work, it was only possible to use a maximum value of p corresponding the size of 34 bits. Evidently, such p values are insufficient to guarantee the security of the proposed protocol because [19] and [10] consider that the size of the $p - value$ for the security of a post-quantum cryptographic protocol based on isogenies of elliptic curves is at least equal to 512 bits. However, this does not invalidate the proof of concept of $\binom{2}{1}$-SIOT.

6 Performance estimate between some OT protocols

In table 2 and figure 6, the number of types of operations by sender and receiver was verified in $\binom{2}{1}$-SIOT and both protocols SIDH-OT and WSW-OT from [26]. Thus, multiplications with scalar and point additions are denoted by *Multi* and *Add*, respectively. Furthermore, calculations for isogenies and pairing are denoted by *Iso* and *Png*, respectively. Therefore, we estimate that the $\binom{2}{1}$-SIOT protocol has slightly better performance than other two protocols.

Table 2. Performance estimation.

Protocol	Sender			Receiver			Png
	Mult	Add	Iso	Mult	Add	Iso	
SIDH-OT	8	4	4	2	2	2	2
WSW-OT	3	3	5	2	1	2	4
SIOT	3	5	3	2	4	2	-

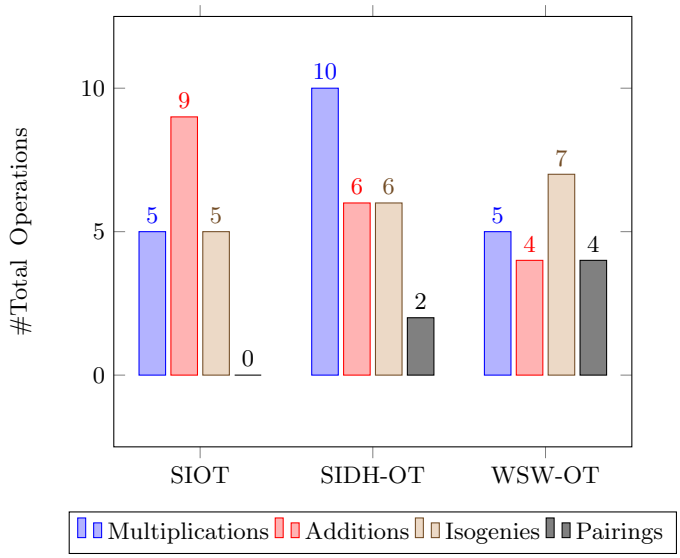


Figure 2. Performance estimation.

7 Conclusion

In this paper, a proposal for a *post quantum* protocol called SIOT is presented. Its security is based on the difficulty of an opponent to calculate isogenies between supersingular elliptic curves and the inspiration of the relative simplicity of the OT protocol of [4] to ensure privacy between the sender and the receiver. With respect to this privacy, it was important to elaborate a theorem, matching a privacy definition of [12] with the computational problems of isogenies of [10], considering a hypothetical scenario between a dishonest sender and an honest receiver and *vice versa*. Finally, an algebraic analysis with *Weil* pairing defined certain necessary conditions so that there were not security and privacy violations in the proposed protocol.

References

1. A.Rostovtsev and A.Stolbunov. Public-key cryptosystem based on isogenies. Cryptology ePrint Archive, Report 2006/145., 2006.

2. Boaz Barak. Oblivious transfer and private information retrieval. Technical report, Princenton University, https://www.cs.princeton.edu/courses/archive/fall07cos433/lec19.pdf, 2007.
3. Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Marie-Hélène Skubiszewska. Practical quantum oblivious transfer. In Joan Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91: Proceedings*, pages 351–366, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.
4. Tung Chou and Claudio Orlandi. The simplest protocol for oblivious transfer. In Kristin Lauter and Francisco Rodríguez-Henríquez, editors, *Progress in Cryptology – LATINCRYPT 2015: 4th International Conference on Cryptology and Information Security in Latin America, Guadalajara, Mexico, August 23-26, 2015, Proceedings*, pages 40–58, Cham, 2015. Springer International Publishing.
5. C. Crepeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 42–52, Oct 1988.
6. Ivan Damgård, Serge Fehr, Carolin Lunemann, Louis Salvail, and Christian Schaffner. Improving the security of quantum protocols via commit-and-open. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, pages 408–427, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
7. A.M.Childs. D.Jao and V. Soukharev. Constructing elliptic curve isogenies in quantum subexponential time., 2014.
8. Shimon Even, Oded Goldreich, and Abraham Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, June 1985.
9. Luca De Feo. *Algorithmes Rapides pour les Tours de Corps Finis et les Isogénies*. PhD thesis, Ecole Polytechnique X, December 2010.
10. Luca De Feo, David Jao, and Jérôme Plût. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. *Journal of Mathematical Cryptology*, 8(3):209–247, 2014.
11. Steven D. Galbraith. *Mathematics of public key cryptography*. Cambridge University Press, Cambridge, 2012.
12. Carmit Hazay and Yehuda Lindell. *Efficient Secure Two - Party Protocols - Techniques and Constructions*. Springer Berlin Heidelberg, 2010.
13. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. *An introduction to mathematical cryptography*. Undergraduate Texts in Mathematics. Springer, New York, second edition, 2014.
14. Yael Tauman Kalai. Smooth projective hashing and two-message oblivious transfer. *EUROCRYPT*, 2005.
15. Raza Ali Kazmi. Cryptography from post-quantum assumptions. Cryptology ePrint Archive, Report 2015/376, 2015. `http://eprint.iacr.org/2015/376`.
16. D. Mayers and L. Salvail. Quantum oblivious transfer is secure against all individual measurements. In *Physics and Computation, 1994. PhysComp '94, Proceedings., Workshop on*, pages 69–77, Nov 1994.
17. Dominic Mayers. Quantum key distribution and string oblivious transfer in noisy channels. In Neal Koblitz, editor, *Advances in Cryptology — CRYPTO '96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings*, pages 343–357, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.
18. Michael O. Rabin. How to exchange secrets with oblivious transfer, 2005. Harvard University Technical Report 81 talr@watson.ibm.com 12955 received 21 Jun 2005.

19. Amir Jalali Mehran Mozaffari Kermani Reza Azarderakhsh, Brian Koziel and David Jao. Neon-sidh: Efficient implementation of supersingular isogeny diffe-hellman key exchange protocol on arm. Cryptology ePrint Archive, Report 2016/669., 2016.
20. Phillip Rogaway. On the role of definitions in and beyond cryptography.
21. Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer, Dordrecht, second edition, 2009.
22. Anton Stolbunov. *Cryptography Shemes based on Isogenies.* PhD thesis, Norwegian University of Science and Technology., 2012.
23. John Tate. Endomorphisms of abelian varieties over finite fields. *Invent. Math.*, 2:134–144, 1966.
24. Dominique Unruh. Universally composable quantum multi-party computation. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 – June 3, 2010. Proceedings*, pages 486–505, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
25. Jacques Vélu. Isogénies entre courbes elliptiques. *C. R. Acad. Sci. Paris Sér. A-B*, 273:A238–A241, 1971.
26. Vanessa Vitse. Simple oblivious transfer protocols compatible with kummer and supersingular isogenies. *Archives-Ouvertes. hal-01981552*, 2019.
27. David Wagner. Technical perspective: Fairness and the coin flip. *Communications of the ACM.*, 59(4):75, April 2016.
28. Lawrence C. Washington. *Elliptic curves - Number Theory and Cryptography.* Taylor & Francis Group. LLC, second edition. edition, 2008.
29. Andrew Chi-Chih Yao. Security of quantum protocols against coherent measurements. In *Proceedings of the Twenty-seventh Annual ACM Symposium on Theory of Computing*, STOC '95, pages 67–75, New York, NY, USA, 1995. ACM.

A Isogenies

In short, isogeny-based cryptography utilizes unique algebraic maps between elliptic curves that satisfy group homomorphism. This original idea introduced by [22] detailed a *Diffie-Hellman* cryptosystem based on the hardness of computing isogenies between ordinary elliptic curves. Nevertheless, [7] developed a *quantum* algorithm that could compute isogenies between ordinary curves in subexponential time. This algorithm uses the fact that the structure of the elliptical group is commutative. Thus, [10] adapted the isogeny-based key exchange protocol to be based on the difficulty of computing isogenies between supersingular elliptic curves, which does not have commutative endomorphism ring.

Definition 4. *Let E_1 and E_2 be elliptic curves over $\mathbb{F}_p$. An isogeny over $\mathbb{F}_p$ is a morphism $\phi\colon E_1 \to E_2$ over $\mathbb{F}_p$ such that $\phi(\mathcal{O}_{E_1}) = \mathcal{O}_{E_2}$ is a group homomorphism. The zero isogeny is the constant map $\phi\colon E_1 \to E_2$ given by $\phi(P) = \mathcal{O}_{E_2}$ for all $P \in E(\overline{\mathbb{F}}_p)$. If there is an isogeny between two elliptic curves E_1 and E_2 then:*

i. *E_1 and E_2 are isogenous;*

ii. $\#E_1(\mathbb{F}_p) = \#E_2(\mathbb{F}_p)$[11]*;*

iii. *E_1 and E_2 have the same $\jmath-$invariant if and only if $E_1 \backsimeq E_2$ over $\bar{\mathbb{F}}_p$ (i.e. exists an isomorfism from E_1 to E_2)*[12]*.*

Definition 5. *Let E_1 and E_2 be elliptic curves over $\mathbb{F}_p$ and $\phi\colon E_1 \to E_2$ over $\mathbb{F}_p$. The degree of a non-zero isogeny is the degree of the morphism. The degree of the zero isogeny is* 0*. If there is an isogeny of degree ℓ between elliptic curves E_1 and E_2 then, they are ℓ-isogenous.*

Definition 6. *Let E_1 and E_2 be elliptic curves over $\mathbb{F}_p$ and $\phi\colon E_1 \to E_2$ an isogeny. Then, the kernel of an isogeny is $ker(\phi) = \{P \in E_1(\bar{\mathbb{F}_p}) : \phi(P) = \mathcal{O}_{E_2}\}$.*

Remark 2. We can denote $E_2 = E_1/ker(\phi)$.

Definition 7. *A non-zero isogeny separable $\phi\colon E_1 \to E_2$ over $\mathbb{F}_p$ of ℓ - degree has $\#ker(\phi) = \ell$.*

Definition 8. *Let $\phi\colon E_1 \to E_2$ and $\hat{\phi}\colon E_2 \to E_3$ be two isogenies with ℓ-degree and $\hat{\ell}$-degree, respectively. Then, their composition is an isogeny $\hat{\phi}(\phi)\colon E_1 \to E_3$ with $(\ell \cdot \hat{\ell})$-degree.*

Proposition 1. *Let E_1 be an elliptic curve over $\mathbb{F}_p$ and $\mathbb{G}$ a finite subgroup of $E_1(\bar{\mathbb{F}}_p)$ that is defined over $\mathbb{F}_p$. Then, there is an unique elliptic curve $E_{\mathbb{G}}$ and a separable isogeny $\phi\colon E_1 \to E_{\mathbb{G}} = E_1/\mathbb{G}$ such that $ker(\phi) = \mathbb{G}$.*

Proof. See Theorem 25.1.6 and Corollary 25.1.7 from [11]. □

B SIDH key exchange

For a better understanding, see the first diagram in figure 3. The sender and the receiver choose randomly two secret integers $r_A \leftarrow \mathbb{Z}/\ell_A^{e_A}\mathbb{Z}$ and $r_B \leftarrow \mathbb{Z}/\ell_B^{e_B}\mathbb{Z}$, respectively. Thus, the kernel $\langle P_A + r_A Q_A\rangle$ from sender has order $\ell_A^{e_A}$ and its secret key is computed as the degree $\ell_A^{e_A}$ isogeny $\phi_A : E_0 \to E_A$, and its pk_A is the isogenous curve E_A together with images $G_A \leftarrow \phi_A(P_B)$ and $H_A \leftarrow \phi_A(Q_B)$.

Similarly, the kernel $\langle P_B + r_B Q_B\rangle$ from receiver has order $\ell_B^{e_B}$ and its secret key is computed as the degree $\ell_B^{e_B}$ isogeny $\phi_B : E_0 \to E_B$, and its pk_B is the isogenous curve E_B together with images $G_B \leftarrow \phi_B(P_A)$ and $H_B \leftarrow \phi_B(Q_A)$. In short, there is a public key exchange, say pk_A and pk_B.

To compute the shared secret k, sender uses its secret integers and receiver's public key to compute the degree ℓ_A isogeny $\phi_A^{'} : E_B \to E_{BA}$ whose kernel is the point $\phi_B(P_A) + r_A\phi_B(Q_A) = \phi_B(P_A + r_A Q_A)$. In the same way, receiver uses its secret integers and sender's public key to compute the degree ℓ_B isogeny $\phi_B^{'} : E_A \to E_{AB}$ whose kernel is the point $\phi_A(P_B) + r_B\phi_A(Q_B) = \phi_A(P_B +$

[11] See [23].

[12] See Theorem 9.3.6 from [11].

$r_B Q_B$). It happens that E_{BA} and E_{AB} are isomorphic. Hence, both sender and receiver can compute a shared secret k, that is, there is the common $\jmath$-invariant $\jmath(E_{BA}) = \jmath(E_{AB})$.

Therefore, $\mathcal{H}(\jmath(E_{BA})) = \mathcal{H}(\jmath(E_{AB})) = k$. After that, the sender computes $c \leftarrow \mathcal{E}_k(m)$ and sends it to the receiver that computes $m \leftarrow \mathcal{D}_k(c)$. More details, see [10]

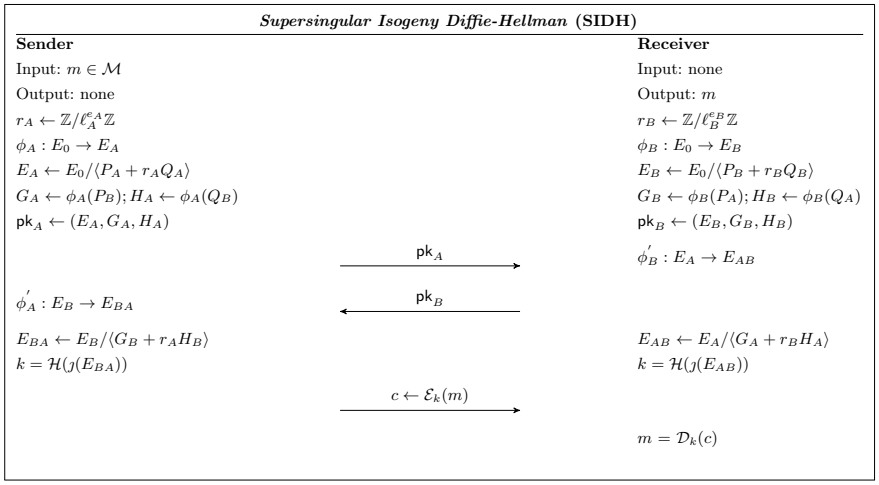


Figure 3. SIDH protocol.

C *Oblivious Transfer* protocol

Oblivious Transfer (OT) is a protocol in which a sender transfers one of many pieces of information to a receiver, but remains oblivious as to what piece has been transferred. The original notion of OT was first proposed by *Michael Rabin* in 1981 [18] in which a sender sends an encrypted message to a receiver and this one could decrypt such message with probability 1/2. After this, [8] presented a general form of OT, named 1-out-of-2 OT, $\binom{2}{1}$ - OT for short, *i.e*, where a sender sends two encrypted messages to a receiver being able to decrypt only one of them.

Many authors have generalized this to $\binom{n}{1}$ - OT where the receiver chooses one message out of n and $\binom{k}{n}$-OT in which the receiver chooses a subset of size k from among n messages. In this work, we will be focused only on $\binom{2}{1}$ - OT.

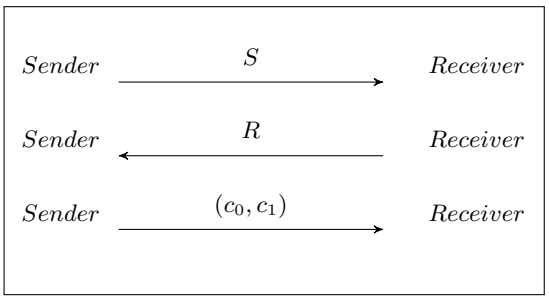
C.1 Protocol $\binom{2}{1}$- OT *Chou-Orlandi*

In this appendix, we see the simplified scheme of the random OT proposed in [4].

Premises

1. The scheme from [4] works in a primitive additive group $(\mathbb{G}, B, \mathbb{F}_p, +)$ of prime order p, generated by base point B;
2. Let s be a safety parameter and $\jmath \in \{0,1\}$. Thus, the hash function $\mathcal{H} : (\mathbb{G} \times \mathbb{G}) \times \mathbb{G} \to \{0,1\}^s$ is used to generate a cryptographic key $k_\jmath$ for use in a symmetric cipher defined by the functions $\mathcal{E}$ (encryption) and $\mathcal{D}$ (decryption), i.e, $c_0 = \mathcal{E}(k_0, m_0)$ and $c_1 = \mathcal{E}(k_1, m_1)$.

3. Abstract view of information exchange from protocol $\binom{2}{1}$- OT *Chou-Orlandi*.



Sender and Receiver

Setup - Sender

1. Sender secretly chooses a value $y \in \mathbb{F}_p$;
2. Sender computes:

$$S = yB \qquad (1)$$
$$T = yS; \qquad (2)$$

3. Sender sends S to Receiver which refuses if $S \notin \mathbb{G}$.

Setup - Receiver

1. Receiver secretly chooses a value $x \in \mathbb{F}_p$;
2. Receiver computes:

$$R = b.S + x.B \qquad (3), \ \textit{where}\, b \in \{0,1\}\, \textit{is chosen by Receiver};$$

3. Receiver sends R to Sender which refuses if $R \notin \mathbb{G}$.

Generation of cryptographic keys $k_\jmath$, $\jmath \in \{0,1\}$.

1. Sender computes $k_\jmath = \mathcal{H}_{(S,R)}(yR - \jmath T)$; (4)
2. Receiver computes $k_b = \mathcal{H}_{(S,R)}(bS + xB)$. (5)

Encryption and Decryption

1. Sender encrypts and sends $c = (c_0, c_1)$ to Receiver. Recalling $c_0 = \mathcal{E}(k_0, M_0)$ and $c_1 = \mathcal{E}(k_1, M_1)$;
2. Receiver decrypts and gets $M_b = \mathcal{D}(k_b, c_\jmath)$, $\jmath \in \{0, 1\}$.

Remark 3. It is verified that a key $k_\jmath$, $j \in \{0, 1\}$, is computed by $\mathcal{H}_{(S,R)}[xyB + (b-j)T]$. Hence, at the end of the protocol if both parts are honest then we have that $k_b = k_\jmath$. In other words, if $\jmath = 0$ then $c = c_0 = 0$ and $k_0 = k_b = \mathcal{H}_{(S,R)}(xyB)$. Otherwise, if $\jmath = 1$ then $c = c_1 = 1$ and $k_1 = k_b = \mathcal{H}_{(S,R)}(xyB)$.

$$
\begin{aligned}
k_\jmath &= yR - \jmath T; \\
&= y(bS + xB) - \jmath T; && from\, equation\ (3) \\
&= byS + xyB - \jmath T; \\
&= bT + xyB - \jmath T; && from\, equations\ (1)\, and\ (2) \\
&= xyB + (b - \jmath)T.
\end{aligned}
$$

Therefore, we can conclude that if the Receiver chooses $b \notin \jmath$, he will not share the secret (cryptographic key) with the Sender.

D Linearly independent points

In this appendix, we present definitions for the understanding of the process that determines the choice of linearly independent points P_A, Q_A, P_B and Q_B in the proposed protocol.

Definition 9 (Frobenius). *Let $E(\mathbb{F}_q)$ be an ellipic curve, and let $E(\mathbb{F}_{q^k})$ be its $\mathbb{F}_{q^k}$-rational extension. The* Frobenius *map is the function $\Phi : E(\mathbb{F}_{q^k}) \to E(\mathbb{F}_{q^k})$ defined by $\Phi(x, y) = (x^q, y^q)$ for any $(x, y) \in E(\mathbb{F}_{q^k})$. Φ^i denotes its i-th self-composition, i.e. for any $P \in E(\mathbb{F}_{q^k})$, $\Phi^i(P) := P$ for $i = 0$, and $\Phi^i(P) = \Phi(\Phi^{i-1}(P))$ for $i > 0$.*

Definition 10 (Trace). *Let $E(\mathbb{F}_q)$ be an ellipic curve, and let $E(\mathbb{F}_{q^k})$ be its $\mathbb{F}_{q^k}$-rational extension. The* trace *map is the function $tr : E(\mathbb{F}_{q^k}) \to E(\mathbb{F}_{q^k})$ defined by $tr(P) = (1/k)\sum_{i=0}^{k-1} \Phi^i(P)$ where $1/k$ denotes the inverse of k mod the order of $E(\mathbb{F}_{q^k})$. In particular, $k = 2$ for a supersingular curve in characteristic $p > 3$, and $tr(P) = (1/2)(P + \Phi(P))$.*

Hence, the trace map is important in that its eigenspaces, if nontrivial, form two linearly independent groups that can be used to sample points P_A, Q_A, P_B, Q_B efficiently. Moreover, the trace definition assumes that $\gcd(k, \#E(\mathbb{F}_{q^k})) = 1$, which may not be the case, especially in the important setting where $\ell_A = 2$. Thus, for this scenario we also define the $quasi - trace$ map:

Definition 11 (Quasi-trace). *Let $E(\mathbb{F}_q)$ be an ellipic curve, and let $E(\mathbb{F}_{q^k})$ be its $\mathbb{F}_{q^k}$-rational extension. The* quasi-trace *map is the function $tr : E(\mathbb{F}_{q^k}) \to E(\mathbb{F}_{q^k})$ defined by $tr(P) = \sum_{i=0}^{k-1} \Phi^i(P)$. In particular, $k = 2$ for a supersingular curve in characteristic $p > 3$, and $tr(P) = P + \Phi(P)$.*

E Possibility of symmetric pairings in the SIOT

Under certain circumstances, it is possible to define a *symmetric* pairing $\hat{e} : E_B[\ell_A^{e_A}] \to F_{p^2}$. We now analyze the condition under which this can happen. In what follows, recall that a distortion map is a linear transformation that maps a curve point to a linearly independent point.

The embedding degree for $E_B[\ell_A^{e_A}]$ is only 1, not 2 as it is for E_0, because E_B is defined over F_q with $q := p^2$, and since $p = (\ell_A^{e_A}\ell_B^{e_B}f)^2 - 1$, it follows that $\#E_B[\ell_A^{e_A}] = (\ell_A^{e_A})^2 \mid q - 1$. Hence a distortion map ψ must map a point $P \in E[\ell_A^{e_A}](F_q)$ to a point $Q \in E[\ell_A^{e_A}](F_q)$ that is linearly independent from P, in which case ψ linearly maps a basis (G_B, H_B) to another basis (G'_B, H'_B).

In particular, all coefficients of ψ in basis (G_B, H_B) must be integers mod $\ell_A^{e_A}$. For such a map to be a distortion map, it must have no eigenvectors (otherwise it would fail to map those points to linearly independent points), so we can simply require the characteristic polynomial to have no roots mod $\ell_A^{e_A}$.

In that case, the map $\psi(uG_B + vH_B) := vG_B - uH_B$ could be a suitable distortion map. Its characteristic polynomial is $\lambda^2 + 1$ which has no roots mod $\ell_A^{e_A}$ for a careful choice of ℓ_A (e.g. $\ell_A = 3$). Now define the modified pairing $\hat{e}(P, Q) := e(P, \psi(Q))$ where $e(\cdot)$ is the *Weil* pairing. Then:

$$\begin{aligned}
\hat{e}(aG_B + bH_B, cG_B + dH_B) &= e(aG_B + bH_B, dG_B - cH_B) \\
&= e(aG_B, -cH_B) \cdot e(bH_B, dG_B) \\
&= e(G_B, H_B)^{-ac-bd}, \\
\hat{e}(cG_B + dH_B, aG_B + bH_B) &= e(cG_B + dH_B, bG_B - aH_B) \\
&= e(cG_B, -aH_B) \cdot e(dH_B, bG_B) \\
&= e(G_B, H_B)^{-ac-bd},
\end{aligned}$$

so this modified pairing is symmetric.

It remains to determine if it is isogeny-equivariant. If it is, a further constraint exists for the coefficients of U and V, namely:

$$\begin{aligned}
\hat{e}(G_B + \lambda U, H_B + \lambda V) &= \hat{e}(G_B, H_B)^{(1+\lambda\alpha)(1+\lambda\delta)} \\
&\cdot\ \hat{e}(H_B, G_B)^{\lambda\beta\lambda\gamma} \\
&= \hat{e}(G_B, H_B)^{(1+\lambda\alpha)(1+\lambda\delta)+\lambda^2\beta\gamma} \\
&= e(G_B, H_B),
\end{aligned}$$

so we also need $(1 + \lambda\alpha)(1 + \lambda\delta) + \lambda^2\beta\gamma = 1 \pmod{\ell_A^{e_A}}$. □

E.1 Taking symmetric pairings into account

Coupling the above constraints $\gamma = -\alpha^2/\beta \pmod{\ell_A^{e_A}}$ and $\delta = -\alpha \pmod{\ell_A^{e_A}}$ with the additional condition $(1 + \lambda\alpha)(1 + \lambda\delta) + \lambda^2\beta\gamma = 1 \pmod{\ell_A^{e_A}}$, we have $(1 + \lambda\alpha)(1 - \lambda\alpha) - \lambda^2\beta\alpha^2/\beta = 1 - 2\lambda^2\alpha^2 = 1 \pmod{\ell_A^{e_A}}$ for any λ, or simply $2\alpha^2 = 0 \pmod{\ell_A^{e_A}}$, which has the solution $\alpha = \alpha_0 \cdot 2^{\lfloor e_A/2 \rfloor}$ for $\ell_A = 2$ and any $0 \leqslant \alpha_0 < 2^{\lceil e_A/2 \rceil}$, or $\alpha = \alpha_0 \cdot \ell_A^{\lceil e_A/2 \rceil}$ for $\ell_A \neq 2$ and any $0 \leqslant \alpha_0 < \ell_A^{\lfloor e_A/2 \rfloor}$. □

F Validating the process of sharing points (U, V)

Now, we are going to verify the sharing of points U and V between Bob and Alice. This is important from the point of view of the correct functionality of the $\binom{2}{1}$-SIOT protocol with regard to the *oblivious* characteristic, *i.e*, in practical terms, points U and V provide the sender to generate two secret keys. Thus, Bob defines points U and V as mentioned in section 2.3. Recall that these points can be written as a linear combination. After that, he sends to Alice one of the pairs (G_B, H_B) or $(G_B - U, H_B - V)$. Obviously, Alice doesn't distinguish[13] which pair of points she received. Thus, upon receipt of $\hat{G}_B$ and $\hat{H}_B$ points from Bob's public key, say $\hat{pk}_B = (E_B, \hat{G}_B, \hat{H}_B)$, Alice defines $\hat{U}$ and $\hat{V}$ yielding $\hat{U} = U$ and $\hat{V} = V$. In other words, Alice and Bob have the assurance that points U and V are being correctly shared between the parties.

Proof.

1. In a first assumption, Alice receives points $\hat{G}_B = (G_B - U)$ and $\hat{H}_B = (H_B - V)$ from Bob. Evidently, she has not any knowledge about points G_B and H_B. Thus, she performs the algebraic development below.

$$
\begin{aligned}
\hat{U} &= \alpha \cdot \hat{G}_B + \beta \cdot \hat{H}_B; \\
\hat{U} &= \alpha \cdot (G_B - U) + \beta \cdot (H_B - V); \\
\hat{U} &= \alpha \cdot G_B - \alpha \cdot U + \beta \cdot H_B - \beta \cdot V; \\
\hat{U} &= \underbrace{\alpha \cdot G_B + \beta \cdot H_B}_{U} - (\alpha \cdot U + \beta \cdot V); \\
\hat{U} &= U - (\alpha \cdot U + \beta \cdot V); \\
\hat{U} &= U - \underbrace{[\alpha \cdot U + \beta \cdot (-\frac{\alpha}{\beta} \cdot U)]}_{0}; \\
\hat{U} &= U.
\end{aligned}
$$

If $\hat{U} = U$ and $V = -(\alpha/\beta)U$, then $\hat{V} = V$. □

2. In this second assumption, Alice receives $\hat{G}_B = G_B$ e $\hat{H}_B = H_B$ points from Bob. Similarly,

[13] See Subsection 3.4, Lemma 1.

$$\hat{U} = \alpha \cdot \hat{G}_B + \beta \cdot \hat{H}_B;$$
$$\hat{U} = \alpha \cdot G_B + \beta \cdot H_B;$$
$$\hat{U} = \underbrace{\alpha \cdot G_B + \beta \cdot H_B}_{U};$$
$$\hat{U} = U.$$

Evidently, in this case, If $\hat{U} = U$ then, $\hat{V} = V$ □
.