SimS

Fouotsa, Tako Boris; Petit, Christophe

DOI:
10.1007/978-3-030-81293-5_15

Document Version
Peer reviewed version

Citation for published version (Harvard):
Fouotsa, TB & Petit, C 2021, SimS: a simplification of SiGamal. in JH Cheon & J-P Tillich (eds), Post-Quantum Cryptography: 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20–22, 2021, Proceedings. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12841 LNCS, Springer, pp. 277-295, 12th International Conference on post-quantum cryptography, PQCrypto 2021, Daejeon, Korea, Republic of, 20/07/21. https://doi.org/10.1007/978-3-030-81293-5_15

InSIDH: a Simplification of SiGamal

Tako Boris Fouotsa[1] and Christophe Petit[2]

[1] Università Degli Studi Roma Tre, Italy
takoboris.fouotsa@uniroma3.it
[2] Université Libre de Bruxelles, Belgium
christophe.f.petit@gmail.com

Abstract. At Asiacrypt 2020, Moriya et al. introduced two new IND-CPA secure supersingular isogeny based Public Key Encryption (PKE) protocols: SiGamal and C-SiGamal. Unlike the PKEs canonically derived from SIKE and CSIDH, the new protocols provide IND-CPA security without the use of random oracles. SiGamal and C-SiGamal are however not IND-CCA secure. Moriya et al. suggested a variant of SiGamal that could be IND-CCA secure, but left its study as an open problem.
In this paper, we revisit the protocols introduced by Moriya et al. First, we show that the SiGamal variant suggested by Moriya et al. for IND-CCA security is, in fact, not IND-CCA secure. Secondly, we propose a new isogeny-based PKE protocol named InSIDH, obtained by simplifying SiGamal. InSIDH has smaller public keys and ciphertexts than (C-)SiGamal and it is more efficient. We prove that InSIDH is IND-CCA secure under CSIDH security assumptions and one Knowledge of Exponent-type assumption we introduce. Interestingly, InSIDH is also much closer to the CSIDH protocol, facilitating a comparison between SiGamal and CSIDH.

Keywords: Post-quantum cryptography · supersingular isogenies · PKE · CSIDH · SiGamal · InSIDH

1 Introduction

The construction of a broad quantum computer would make the nowadays widely used public PKE schemes insecure, namely RSA [27], ECC [19] and their derivatives. As a response to the considerable progress in constructing quantum computers, NIST launched a standardization process for post-quantum secure protocols in December 2016 [24].

The idea of using the isogeny computation problem as hard problem in cryptography is due to J. M. Couveignes in 1997 [9]. About a decade later, many isogeny based schemes surged, among which a Key Exchange protocol by Rostovtsev and Stolbunov [29] based on ordinary isogenies, a hash function by Charles, Goren and Lauter[7] and another key exchange protocol (SIDH) by Jao and De Feo [18] based supersingular isogenies. The submission of SIKE [17] (which is a Key Encapsulation Mechanism based on SIDH) to the NIST standardization process

marked the starting point of a more active research in isogeny based cryptography. Isogeny based protocols are in general based on the assumption that given two isogenous curves E and E', it is difficult to compute an isogeny from E to E'. Even though they are not the most efficient candidates for post quantum cryptography protocols, they provide the shortest public keys and ciphertexts.

PKE schemes based on isogeny problems include SIKE, but also CSIDH [5], SÉTA [12] and more recently SiGamal and C-SiGamal [22]. SÉTA and the PKEs canonically derived from the key exchange protocols SIDH and CSIDH require hash functions and/or generic transformations such as the Fujisaki-Okamoto [14] or OAEP [1] to provide IND-CPA security ([12, §2.4],[17, §1.4], [22, §3.3]). This motivated Moriya, Onuki and Tagaki to introduce the SiGamal [22, §5] and C-SiGamal [22, §6] PKE schemes derived from CSIDH. SiGamal and C-SiGamal provide IND-CPA security under new assumptions they introduce. The authors noticed that neither SiGamal nor C-SiGamal is IND-CCA secure. In Remark 7 of [22], they suggest a slightly modified version of SiGamal that from their point of view could be IND-CCA secure, but they left its study as open problem.

Contributions. In this paper, we prove that the variant of SiGamal suggested by Moriya et al. in Remark 7 of their paper is not IND-CCA secure by exhibiting a simple and concrete attack. We then modify SiGamal to thwart this attack, and obtain a new isogeny-based PKE scheme called InSIDH. We prove that InSIDH is IND-CPA secure relying on CSIDH security assumptions (Assumption 2). This is a considerable improvement on SiGamal whose IND-CPA security relies on new assumptions. We introduce a "knowledge of Exponent" type assumption (Assumption 3) under which we prove that InSIDH is IND-CCA secure. This assumption may have some other applications in isogeny based cryptography.
We adapt the Magma code for SiGamal [21] to run a proof of concept implementation of InSIDH using the SiGamal primes p_{128} and p_{256}. For the prime p_{128}, InSIDH is about 1.13x faster than SiGamal and about 1.19x faster than C-SiGamal. For the prime p_{256}, we get a 1.07x speedup when compared to SiGamal and a 1.21x speedup when compared to C-SiGamal.
For the same set of parameters, InSIDH has smaller private keys, public keys and ciphertexts compared to SiGamal and C-SiGamal. InSIDH is simple, seats between SiGamal and CSIDH, helps to better understand the relation between SiGamal and CSIDH while providing IND-CCA security and being more efficient compared to SiGamal. Table 3 best summarizes our contributions.

Outline. The remaining of this paper is organized as follows: in Section 2, we recall the security definitions for PKE schemes, the main ideas of the class group action and the CSIDH key exchange protocol. In section 3, we present the SiGamal PKE scheme and we show that the variant suggested in [22, Remark 7] is not IND-CCA secure. Section 4 is devoted to InSIDH and its security arguments. In section 5 we present the outcome of a proof-of-concept implementation and compare InSIDH to CSIDH and (C-)SiGamal in Section 6. We conclude the paper in Section 7.

2 Preliminaries

2.1 Public key encryption

We recall standard security definitions related to public key encryption.

Definition 1 (PKE). *A Public Key Encryption scheme $\mathcal{P}_\lambda$ is a triple of PPT algorithms (*Key Generation*,* Encryption*,* Decryption*) that satisfy the following.*

1. *Given a security parameter λ as input, the key generation algorithm* Key Generation *outputs a public key pk, a private key sk and a plaintext space $\mathcal{M}$.*
2. *Given a plaintext $\mu \in \mathcal{M}$ and a public key pk as inputs, the encryption algorithm* Encryption *outputs a ciphertext $\mathsf{c} = \mathsf{Encryption}_{pk}(\mu)$.*
3. *Given a ciphertext c and sk as inputs, the decryption algorithm* Decryption *outputs a plain text $= \mathsf{Decryption}_{sk}(\mathsf{c})$.*

Definition 2 (Correctness). *A PKE scheme $\mathcal{P}_\lambda$ is correct if for any pair of keys (pk, sk) and for every plaintext $\mu \in \mathcal{M}$,*

$$\mathsf{Decryption}_{sk}\left(\mathsf{Encryption}_{pk}(\mu)\right) = \mu.$$

Definition 3 (IND-CPA secure). *A PKE scheme $\mathcal{P}_\lambda$ is IND-CPA secure if for every PPT adversary $\mathcal{A}$,*

$$Pr\left[b = b^* \left| \begin{array}{l} (pk, sk) \leftarrow \mathsf{Key\ Generation}(\lambda), \mu_0, \mu_1 \leftarrow \mathcal{M}, \\ b \xleftarrow{\$} \{0,1\}, \mathsf{c} \leftarrow \mathsf{Encryption}_{pk}(\mu_b), b^* \leftarrow \mathcal{A}(pk, \mathsf{c}) \end{array} \right.\right] = \frac{1}{2} + \mathsf{negl}(\lambda).$$

Definition 4 (IND-CCA secure). *A PKE scheme $\mathcal{P}_\lambda$ is IND-CCA secure if for every PPT adversary $\mathcal{A}$,*

$$Pr\left[b = b^* \left| \begin{array}{l} (pk, sk) \leftarrow \mathsf{Key\ Generation}(\lambda), \mu_0, \mu_1 \leftarrow \mathcal{A}^{O(\cdot)}(pk, \mathcal{M}), \\ b \xleftarrow{\$} \{0,1\}, \mathsf{c} \leftarrow \mathsf{Encryption}_{pk}(\mu_b), b^* \leftarrow \mathcal{A}^{O(\cdot)}(pk, \mathsf{c}) \end{array} \right.\right] = \frac{1}{2} + \mathsf{negl}(\lambda),$$

where $O(\cdot)$ is a decryption oracle that when given a ciphertext $\mathsf{c}' \neq \mathsf{c}$, outputs $\mathsf{Decryption}_{sk}(\mathsf{c}')$ or $\perp$ if the ciphertext c' is invalid.

2.2 Class group action on supersingular curves defined over $\mathbb{F}_p$

We refer to [28,30] for general mathematical background on supersingular elliptic curves and isogenies, to [5,13] for supersingular elliptic curves defined over $\mathbb{F}_p$ and their $\mathbb{F}_p$-endomorphism ring, and to [8,26] for isogenies between Montgomery curves.

Let $p \equiv 3 \mod 4$ be a prime greater than 3. The equation $By^2 = x^3 + Ax^2 + x$ where $B \in \mathbb{F}_p^*$ and $A \in \mathbb{F}_p \setminus \{\pm 2\}$ defines a Montgomery elliptic curve E over $\mathbb{F}_p$. The j-invariant of E is defined as $j = \frac{256(A^2-3)^3}{A^2-4}$. Two elliptic curves are isomorphic if and only if they have the same j-invariant. The curve $E : By^2 = x^3 + Ax^2 + x$ is isomorphic (over $\mathbb{F}_p$) to the curve defined by the equation

$y^2 = x^3 + Ax^2 + x$ (resp. $-y^2 = x^3 + Ax^2 + x$) when B is a square in $\mathbb{F}_p$ (resp. B is not a square in $\mathbb{F}_p$). The curve E is said to be supersingular if $\sharp E(\mathbb{F}_p) \equiv 1 \mod p$, otherwise E is said to be ordinary. If E is a supersingular curve defined over $\mathbb{F}_p$, then $\sharp E(\mathbb{F}_p) = p + 1$. All the elliptic curves we consider in this paper are supersingular curves defined by an equation of the form $y^2 = x^3 + Ax^2 + x$ where $A \in \mathbb{F}_p$ is called the Montgomery coefficient of the curve.
The endomorphism ring of a supersingular curve is a maximal order in a quaternion algebra. Since E is defined over $\mathbb{F}_p$, then some of its endomorphisms are also defined over $\mathbb{F}_p$. Let π be the Frobenius endomorphism of E. The $\mathbb{F}_p$-endomorphism ring $\mathcal{O}$ of E is isomorphic to either $\mathbb{Z}[\pi]$ or $\mathbb{Z}[\frac{1+\pi}{2}]$ [13]. As in the ordinary case, the class group $\mathsf{cl}(\mathcal{O})$ of $\mathcal{O}$ acts freely and transitively on the set $\mathcal{E}\ell\ell_p(\mathcal{O})$ of supersingular elliptic curves defined over $\mathbb{F}_p$ and having $\mathbb{F}_p$-endomorphism ring $\mathcal{O}$. We have the following theorem.

Theorem 1. *[5, Theorem 7] Let $\mathcal{O}$ be an be an order in an imaginary quadratic field such that $\mathcal{E}\ell\ell_p(\mathcal{O})$ is non empty. The ideal class group $\mathsf{cl}(\mathcal{O})$ acts freely and transitively on the set $\mathcal{E}\ell\ell_p(\mathcal{O})$ via the map*

$$\begin{array}{rcl} \mathsf{cl}(\mathcal{O}) \times \mathcal{E}\ell\ell_p(\mathcal{O}) \to & \mathcal{E}\ell\ell_p(\mathcal{O}) \\ ([\mathfrak{a}], E) & \mapsto [\mathfrak{a}]E = E/E[\mathfrak{a}], \end{array}$$

where $\mathfrak{a}$ is an integral ideal of $\mathcal{O}$ and $E[\mathfrak{a}] = \cap_{\alpha \in \mathfrak{a}} \ker \alpha$.

From now on, we will consider the quadratic order $\mathbb{Z}[\pi]$ and the action of its class group $\mathsf{cl}(\mathbb{Z}[\pi])$ on the set $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi])$. We represent isomorphism classes of curves in $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi])$ using the Montgomery coefficient A [3, Proposition 3].

The efficiency of the computation of an isogeny with known kernel essentially depends on the smoothness of its degree. In [5], the authors work with a prime p of the form $p = 4\ell_1 \cdots \ell_n - 1$. This implies that for $i \in \{1, \cdots, n\}$, $\left(\frac{-p}{\ell_i}\right) = 1$ and by the Kummer decomposition theorem [20], $(\ell_i) = \mathfrak{l}_i\overline{\mathfrak{l}_i}$ in $\mathsf{cl}(\mathbb{Z}[\pi])$, where $\mathfrak{l}_i = (\ell_i, \pi - 1)$ and $\overline{\mathfrak{l}_i} = (\ell_i, \pi + 1)$ are integral ideals of prime norm ℓ_i. It follows that $[\mathfrak{l}_i][\overline{\mathfrak{l}_i}] = [\ell_i] = [1]$ in $\mathsf{cl}(\mathbb{Z}[\pi])$, hence $[\mathfrak{l}_i]^{-1} = [\overline{\mathfrak{l}_i}]$. Since the primes ℓ_i are small, then the action of the ideal classes $[\mathfrak{l}_i]$ and $[\mathfrak{l}_i]^{-1}$ can be computed efficiently using Vélu formulas for Montgomery curves [8,26]. In reality, the kernel of the isogeny corresponding to the action of the prime ideal $\mathfrak{l}_i = (\ell_i, \pi - 1)$ is generated by a point $P \in E(\mathbb{F}_p)$ of order ℓ_i, while that of the isogeny corresponding to the action of $\mathfrak{l}_i^{-1} = (\ell_i, \pi + 1)$ is a point $P' \in E(\mathbb{F}_{p^2}) \setminus E(\mathbb{F}_p)$ of order ℓ_i such that $\pi(P') = -P'$. The computation of the action of an ideal class $\prod[\mathfrak{l}_i]^{e_i}$ where $(e_1, \cdots, e_n) \in \{-m, \cdots, m\}^n$ can be done efficiently by composing the actions of the ideal classes $[\mathfrak{l}_i]$ or $[\mathfrak{l}_i]^{-1}$ depending on the signs of the exponents e_i. Since the prime ideals $\mathfrak{l}_i$ are fixed, then the vector $(e_1, \cdots, e_n)$ is used to represent the ideal class $\prod[\mathfrak{l}_i]^{e_i}$. From the discussion in [5, §7.1], m is chosen to be the least positive integer such that

$$(2m + 1)^n \geq |\mathsf{cl}(\mathbb{Z}[\pi])| \approx \sqrt{p}.$$

2.3 CSIDH

CSIDH [5] stands for Commutative Supersingular Isogeny Diffie-Hellman and is a Diffie-Hellman type key exchange protocol. The base group in Diffie-Hellman protocol is replaced by the unstructured set $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi])$ and the exponentiation is replaced by the class group action of $\mathsf{cl}(\mathbb{Z}[\pi])$ on $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi])$. Concretely, CSIDH is designed as follows.

Setup. Let $p = 4\ell_1 \cdots \ell_n - 1$ be a prime where $\ell_1, \cdots, \ell_n$ are small distinct odd primes. The prime p and the supersingular elliptic curve $E_0 : y^2 = x^3 + x$ defined over $\mathbb{F}_p$ with $\mathbb{F}_p$-endomorphism $\mathbb{Z}[\pi]$ are the public parameters.

Key Generation. The private key is an n-tuple $e = (e_1, \cdots, e_n)$ of uniformly random integers sampled from a range $\{-m, \cdots, m\}$. This private key represents an ideal class $[\mathfrak{a}] = \prod[\mathfrak{l}_i]^{e_i} \in \mathsf{cl}(\mathbb{Z}[\pi])$. The public key is the Montgomery coefficient $A \in \mathbb{F}_p$ of the curve $[\mathfrak{a}]E_0 : y^2 = x^3 + Ax^2 + x$ obtained by applying the action of $[\mathfrak{a}]$ on E_0.

KeyExchange Suppose Alice and Bob have successfully computed pairs of private and public key (e, A) and (e', B) respectively. Upon receiving Bob's public key $B \in \mathbb{F}_p \setminus \{\pm 2\}$, Alice verifies that the elliptic curve $E_B : y^2 = x^3 + Bx^2 + x$ is a supersingular curve, then applies the action of the ideal class corresponding to her secret key $e = (e_1, \cdots, e_n)$ to E_B to compute the curve $[\mathfrak{a}]E_B = [\mathfrak{a}][\mathfrak{b}]E_0$. Bob does analogously with his own secret key $e' = (e'_1, \cdots, e'_n)$ and Alice's public key $A \in \mathbb{F}_p \setminus \{\pm 2\}$ to compute the curve $[\mathfrak{b}]E_A = [\mathfrak{b}][\mathfrak{a}]E_0$. The shared secret is the Montgomery coefficient S of the common secret curve $[\mathfrak{a}][\mathfrak{b}]E_0 = [\mathfrak{b}][\mathfrak{a}]E_0$.

The security of the CSIDH key exchange protocol relies on the following assumptions.
Let λ be the security parameter and let $p = 4\ell_1 \cdots \ell_n - 1$ be a prime where $\ell_1, \cdots, \ell_n$ are small distinct odd primes. Let E_0 be the supersingular elliptic curve $y^2 = x^3 + x$ defined over $\mathbb{F}_p$, let $[\mathfrak{a}]$, $[\mathfrak{b}]$ and $[\mathfrak{c}]$ be uniformly random ideal classes in $\mathsf{cl}(\mathbb{Z}[\pi])$.

Assumption 1 *The CSSICDH (Commutative Supersingular Isogeny Computational Diffie-Hellman) assumption holds if for any probabilistic polynomial time (PPT) algorithm $\mathcal{A}$,*

$$Pr[E = [\mathfrak{b}][\mathfrak{a}]E_0 \mid E = \mathcal{A}(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0)] < \mathsf{negl}(\lambda).$$

Assumption 2 *The CSSIDDH (Commutative Supersingular Isogeny Decisional Diffie-Hellman) assumption holds if for any PPT algorithm $\mathcal{A}$,*

$$Pr\left[b = b^* \;\middle|\; \begin{array}{l} [\mathfrak{a}], [\mathfrak{b}], [\mathfrak{c}] \leftarrow \mathsf{cl}(\mathbb{Z}[\pi]), b \xleftarrow{\$} \{0,1\}, \\ F_0 := [\mathfrak{b}][\mathfrak{a}]E_0, F_1 = [\mathfrak{c}]E_0, \\ b^* \leftarrow \mathcal{A}(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0, F_b) \end{array}\right] = \frac{1}{2} + \mathsf{negl}(\lambda).$$

In [6], Castryck et al. show that Assumption 2 does not hold for primes $p \equiv 1 \bmod 4$. This does not affect primes $p \equiv 3 \bmod 4$, which are used in CSIDH, SiGamal and in our proposal InSIDH.

An IND-CPA insecure PKE from CSIDH. A PKE scheme can be canonically derived from a key exchange protocol. For the case of CSIDH, this PKE scheme is sketched as follows. Suppose that Alice has successfully computed her key pair (e, A). In order to encrypt a message $\mathsf{m} \in \{0,1\}^{\lceil \log p \rceil}$, Bob computes a random key pair (e', B) and the binary representation S_{01} of the corresponding shared secret S. He sends $(B, \mathsf{c} = S_{01} \oplus \mathsf{m})$ to Alice as the ciphertext. For the decryption, Alice computes the shared secret S and its binary representation S_{01}, then recovers $\mathsf{m} = S_{01} \oplus \mathsf{c}$. In the comparison in Section 6, the term CSIDHpke will be used to refer to the previous PKE each time the precision is needed.

The above PKE scheme is not IND-CPA secure. In fact, given two distinct plaintexts m_0 and m_1, if (B, c) is a ciphertext for m_i, then $S_{01}^i = \mathsf{c} \oplus \mathsf{m}_i$ is the binary representation of the Montgomery coefficient of a supersingular curve while $S_{01}^{1-i} = \mathsf{c} \oplus \mathsf{m}_{1-i}$ is that of an ordinary curve with overwhelming probability. Hence an adversary can efficiently guess if the ciphertext (B, c) is that of m_0 or m_1. In practice, a hash function h is used to mask the supersingular property of the shared secret S, the ciphertext becomes $(B, \mathsf{c} = h(S_{01}) \oplus \mathsf{m})$.

3 Another look at SiGamal protocol

3.1 SiGamal protocol and variants

The SiGamal PKE scheme can be summarized as follows.

Key Generation. Let $p = 2^r \ell_1 \cdots \ell_n - 1$ be a prime such that $\ell_1, \cdots, \ell_n$ are small distinct odd primes. Let E_0 be the elliptic curve $y^2 = x^3 + x$ and let $P_0 \in E(\mathbb{F}_p)$ be a point of order 2^r. Alice takes a random integral ideal $\mathfrak{a} = (\alpha)\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$ where α is a uniformly random element of $\mathbb{Z}_{2^r}^\times$, computes $E_1 := [\mathfrak{a}]E_0$ and $P_1 := \mathfrak{a}P_0$. Her public key is $(E_1, x(P_1))$ and her private key is $(\alpha, e_1, \cdots, e_n)$. Let $\mathbb{Z}_{2^{r-2}} = \mathbb{Z}/2^{r-2}\mathbb{Z}$ be the message space.

Encryption. Let $\mathsf{m} \in \mathbb{Z}_{2^{r-2}}$ be a plaintext, Bob embeds m in $\mathbb{Z}_{2^r}^\times$ via $\mathsf{m} \mapsto M = 2\mathsf{m} + 1$. Bob takes a random integral ideal class $\mathfrak{b} = (\beta)\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$ where β is a uniformly random element of $\mathbb{Z}_{2^r}^\times$. Next, he computes $[M]P_1$, $E_3 = [\mathfrak{b}]E_0$, $P_3 := \mathfrak{b}P_0$, $E_4 = [\mathfrak{b}]E_1$ and $P_4 := \mathfrak{b}([M]P_1)$. He sends $(E_3, x(P_3), E_4, x(P_4))$ to Alice as the ciphertext.

Decryption. Upon receiving $(E_3, x(P_3), E_4, x(P_4))$, Alice computes $\mathfrak{a}P_3$ and solves a discrete logarithm instance between P_4 and $\mathfrak{a}P_3$ using the Pohlig-Hellman algorithm [25]. Let $M \in \mathbb{Z}_{2^r}^\times$ be the solution of this computation. If $2^{r-1} < M$, then Alice changes M to $2^r - M$. She computes the plaintext $\mathsf{m} = (M-1)/2$.

In C-SiGamal, a compressed version of SiGamal, one replaces the point $\mathfrak{a}\mathfrak{b}P_0$ by a distinguished point $P_{E_4} \in E_4$ of order 2^r, which then does not need to be transmitted. The scheme integrates an algorithm that canonically computes a distinguished point of order 2^r on a given supersingular curve defined over $\mathbb{F}_p$

where $p = 2^r l_1 \cdots l_n - 1$. We refer to [22] for more details on the SiGamal and C-SiGamal.

Moriya et al. prove that SiGamal and C-SiGamal are IND-CPA secure relying on two assumptions they introduce. However, they point out that SiGamal is not IND-CCA secure since one can efficiently compute a valid encryption of $3\mathsf{m}+1$ from a valid encryption of m. Indeed, given $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [\mathfrak{b}]E_1, [2\mathsf{m}+1]\mathfrak{b}P_1)$ one easily computes $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [\mathfrak{b}]E_1, [3][2\mathsf{m}+1]\mathfrak{b}P_1) = ([\mathfrak{b}]E_0, \mathfrak{b}P_0, [\mathfrak{b}]E_1, [2(3\mathsf{m}+1)+1]\mathfrak{b}P_1)$. A similar argument applies for C-SiGamal as well.
As a remedy, Moriya et al. suggest to omit the curve $[\mathfrak{b}]E_1$ in the ciphertext (see [22, Remark 7]). We now show that this variant is still vulnerable to IND-CCA attacks.

3.2 An IND-CCA attack on Moriya et al.'s variant

In this version of SiGamal, a ciphertext for m is of the form $([\mathfrak{b}]E_0, \mathfrak{b}P_0, [2\mathsf{m}+1]\mathfrak{b}P_1)$ and the decryption process is identical to that of the original SiGamal. We prove the following lemma.

Lemma 1. *Let (m, c) be a pair of plaintext-ciphertext, and let m' be any other plaintext. One can compute a valid ciphertext for m' in polynomial time.*

Proof. Write $\mathsf{c} = ([\mathfrak{b}]E_0, \mathfrak{b}P_0, [2\mathsf{m}+1]\mathfrak{b}P_1)$. Since $2\mathsf{m}+1, 2\mathsf{m}'+1 \in \mathbb{Z}_{2^r}^\times$, then $\alpha = (2\mathsf{m}+1)(2\mathsf{m}'+1)^{-1} \in \mathbb{Z}_{2^r}^\times$. Since the curve $[\mathfrak{b}]E_0$ and its point $\mathfrak{b}P_0$ are available in c, then the ciphertext $\mathsf{c}' = ([\mathfrak{b}]E_0, [\alpha]\mathfrak{b}P_0, [2\mathsf{m}+1]\mathfrak{b}P_1)$ can be efficiently computed at the cost of a point multiplication by α.

We now show that c' is a valid encryption of m'. To decrypt c', Alice computes $[\mathfrak{a}][\mathfrak{b}]E_0$ and $\mathfrak{a}([\alpha]\mathfrak{b}P_0) = [\alpha]\mathfrak{a}\mathfrak{b}P_0$, then she solves a discrete logarithm problem between $[2\mathsf{m}+1]\mathfrak{b}P_1 = [2\mathsf{m}+1]\mathfrak{a}\mathfrak{b}P_0$ and $[\alpha]\mathfrak{a}\mathfrak{b}P_0$. We have

$$[2\mathsf{m}+1]\mathfrak{a}\mathfrak{b}P_0 = [\alpha^{-1}(2\mathsf{m}+1)][\alpha]\mathfrak{a}\mathfrak{b}P_0.$$

Hence the solution of the discrete logarithm problem is

$$M' = \pm\alpha^{-1}(2\mathsf{m}+1) = \pm(2\mathsf{m}'+1)(2\mathsf{m}+1)^{-1}(2\mathsf{m}+1) = \pm(2\mathsf{m}'+1).$$

It follows that the corresponding plaintext (after changing M' to $2^r - M'$ when necessary) is $(M'-1)/2 = \mathsf{m}'$.

Corollary 1. *The variant of SiGamal suggested by Moriya et al. in [22, Remark 7] is not IND-CCA secure.*

4 InSIDH (Intermediary Supersingular Isogeny Diffie-Hellman)

We now introduce a new protocol that resists the previous attack. Its name highlights the fact that our scheme is intermediary between CSIDH and SiGamal, and the fact that it provides additional insight on the relationships between CSIDH and SiGamal.

4.1 Overview

We observe that the attack presented in the previous section is effective because the ciphertext contains the curve $\mathfrak{b}E_0$ and its 2^r-torsion points $\mathfrak{b}P_0$.

InSIDH is obtained by adjusting SiGamal in such a way that when a curve is part of the ciphertext, then none of its points is, and the other way around. In order to achieve this, we replace the point $\mathfrak{a}\mathfrak{b}P_0$ in the the (C)SiGamal protocol by a canonical point $P_{E_4} \in E_4 = [\mathfrak{a}][\mathfrak{b}]E_0$. More concretely, in InSIDH, Alice's secret key is an ideal class $[\mathfrak{a}]$, and her public key is the curve $E_1 = [\mathfrak{a}]E_0$. To encrypt a message m, Bob chooses a uniformly random ideal class group $[\mathfrak{b}]$, he computes $E_3 = [\mathfrak{b}]E_0$, $E_4 = [\mathfrak{b}]E_1$ and he then canonically computes a point $P_{E_4} \in E_4(\mathbb{F}_p)$ of smooth order $2^r|p+1$. He sends E_3 and $P_4 = [2\mathsf{m}+1]P_{E_4}$ to Alice. In order to recover m, Alice computes $E_4 = [\mathfrak{a}]E_3$ and P_{E_4}, then solves a discrete logarithm instance in a group of order 2^r using the Pohlig-Hellman algorithm. Figure 1 pictures the scheme.

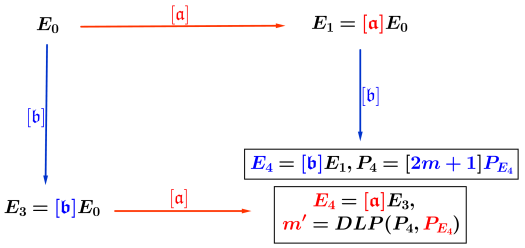

Fig. 1: InSIDH scheme. The elements in black are public, while those in blue are known only by Bob and those in red only by Alice.

The IND-CCA attack presented in Section 3.2 is no more feasible in InSIDH since no point of the curve E_3 nor the curve E_4 are part of the ciphertexts.

4.2 The InSIDH public key encryption protocol

Now let us concretely describe the key generation, encryption and decryption processes. We use the Algorithm 1 to canonically compute the point $P_E \in E(\mathbb{F}_p)$ of order $2^r|p+1$.

Before we describe the protocol, let us notice that revealing P_4 or it x-coordinate may leak too much information about the curve E_4. To avoid this, we make use of a randomizing function $f_E : \mathbb{F}_p \to \mathbb{F}_p$, indexed by supersingular curves defined over $\mathbb{F}_p$, satisfying the following conditions:

- f_E is bijective, f_E and its inverse $g_E = f_E^{-1}$ can be efficiently computed when E is given;
- for every element $x \in \mathbb{F}_p$, an adversary having no access to x and E can not distinguish $f_E(x)$ from a random element of $\mathbb{F}_p$;
- for every element $x \in \mathbb{F}_p$, for every non identical rational function $R \in \mathbb{F}_p(X)$, an adversary having no access to x and E can not compute $f_E(R(x))$ from $f_E(x)$.

Example 1. In the proof of concept implementation in Section 5, we use the function $f_E : x \mapsto x'$ where $bits(x') = bits(x) \oplus bits(A_E)$ and $bits(\cdot)$ takes an element in $\mathbb{F}_p$ and returns its binary representation.

Having such a function, InSIDH is designed as follows.

Key Generation: Let $p = 2^r \ell_1 \cdots \ell_n - 1$ be a prime such that $\ell_1, \cdots, \ell_n$ are small distinct odd primes and $\lambda + 2 \leq r \leq \frac{1}{2} \log p$ where λ is the security parameter. Let E_0 be the elliptic curve $y^2 = x^3 + x$. Alice takes a random ideal class $[\mathfrak{a}] \in \mathsf{cl}(\mathbb{Z}[\pi])$, computes $E_1 := [\mathfrak{a}]E_0$. Her public key is E_1 and her private key is $[\mathfrak{a}]$. The plaintext space is the set $\mathcal{M} = \mathbb{Z}_{2^{r-2}}$.

Encryption: Let $\mathsf{m} \in \mathbb{Z}_{2^{r-2}}$ be a plaintext, Bob embeds m in $\mathbb{Z}_{2^r}^\times$ via $\mathsf{m} \mapsto 2\mathsf{m}+1$. Bob takes a random ideal class $[\mathfrak{b}] \in \mathsf{cl}(\mathbb{Z}[\pi]$ and computes $E_3 = [\mathfrak{b}]E_0$, $E_4 = [\mathfrak{b}]E_1$ and $P_4 = [2\mathsf{m}+1]P_{E_4}$. He sends $(E_3, x' = f_{E_4}(x(P_4)))$ to Alice as the ciphertext.

Decryption: Upon receiving (E_3, x'), Alice verifies that E_3 is a supersingular curve, computes $E_4 = [\mathfrak{a}]E_3$ and P_{E_4}. If $g_{E_4}(x')$ is not the x-coordinate of a 2^r-torsion point on the curve E_4, then Alice aborts. She solves the discrete logarithm instance between $P_4 = (g_{E_4}(x'), -)$ and P_{E_4} using the Pohlig-Hellman algorithm. Let $M \in \mathbb{Z}_{2^r}^\times$ be the solution of this computation. If $2^{r-1} < M$, then Alice changes M to $2^r - M$. She computes the plaintext $(M-1)/2$.

Theorem 2. *InSIDH is correct.*

Proof. As in CSIDH, the Montgomery coefficients of the curves $[\mathfrak{a}][\mathfrak{b}]E_0$ and $[\mathfrak{b}][\mathfrak{a}]E_0$ are equal. Therefore Alice and Bob obtain the same distinguish point P_{E_4}. Since the points P_{E_4} and $P_4 = [2\mathsf{m}+1]P_{E_4}$ have order 2^r, then the Pohlig-Hellman algorithm can be implemented on their x-coordinates $x(P_4) = g_{E_4}(x')$ and $x(P_{E_4})$ only to recover $M \equiv \pm(2\mathsf{m}+1) \bmod 2^r$. Since $\mathsf{m} \in \mathbb{Z}_{2^{r-2}}$, then $2\mathsf{m}+1 < 2^{r-1}$. Alice changes M to $2^r - M$ if $2^{r-1} < M$, then she computes the plaintext $(M-1)/2 = \mathsf{m}$.

Remark 1. Instantiating InSIDH with SIDH would lead to a PKE scheme which is not IND-CCA secure because SIDH is vulnerable to adaptive attacks [15].

4.3 Security arguments

We prove that the IND-CPA security of InSIDH relies on Assumption 2. We also prove that InSIDH is IND-CCA secure under a Knowledge of exponent-type assumption which we introduce.

Theorem 3. *If Assumption 2 holds, then InSIDH is IND-CPA secure.*

Proof. We adapt the proof of [10, Theorem 1] to our setting. Let us suppose that InSIDH is not IND-CPA secure, then there exists a PPT adversary $\mathcal{A}$ that

can successfully distinguish whether a given ciphertext (E_3, x') was encrypted from a plaintext m_0 or m_1 with a non negligible advantage γ. We will use $\mathcal{A}$ to construct a PPT CSSIDDH solver $\mathcal{A}'$ that breaks Assumption 2.

Let $(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0, E)$ be a tuple given to us as a CSSIDDH instance input. Our goal is to decide if this is a **correct** tuple ($[\mathfrak{a}][\mathfrak{b}]E_0 = E$) or a **bad** tuple ($[\mathfrak{a}][\mathfrak{b}]E_0 \neq E$).
Let T be the following two-steps test.

- **Simulation**. One simulates an InSIDH instance using $(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0, E)$. Concretely, one computes P_E, secretly chooses a random bit $b \in \{0, 1\}$ and returns the ciphertext $\mathsf{c} = ([\mathfrak{b}]E_0, f_E(x([2\mathsf{m}_b + 1]P_E)))$.
- **Query** $\mathcal{A}$. One queries $\mathcal{A}$ with the ciphertext c and gets a response b'. The result of the test T is 1 if $b = b'$ and 0 if $b \neq b'$.

Now we distinguish two cases.

Case 1: the adversary $\mathcal{A}$ can detect invalid ciphertexts by returning an error message. Here we run the test T once.
If the result of the query step is an error message instead of a bit, then c is an invalid ciphertext. Hence $E \neq [\mathfrak{a}][\mathfrak{b}]E_0$.
If in the query step $\mathcal{A}$ returns a bit b', then c is a valid ciphertext. Hence $[\mathfrak{a}][\mathfrak{b}]E_0 = E$.
We therefore construct our CSSIDDH solver $\mathcal{A}'$ as follows: if the query step result is an error message, $\mathcal{A}'$ returns bad; if it is a bit, $\mathcal{A}'$ returns $\mathsf{correct}$.

Case 2: the adversary $\mathcal{A}$ cannot detect invalid ciphertexts. Here the query step result will always be a bit b'. The CSSIDDH solver $\mathcal{A}'$ repeats the test T and studies the proportion $\Pr_T(1)$ of 1's obtained.
Suppose that $(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0, E)$ is a correct tuple, then all the ciphertexts c computed in the simulation steps are valid, hence the adversary $\mathcal{A}$ has the same advantage as in an actual attack. Therefore,

$$\Pr_T(1) = \frac{1}{2} + \gamma.$$

On the other hand, let suppose that $(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0, E)$ is a bad tuple. Then $[\mathfrak{a}][\mathfrak{b}]E_0 \neq E$ and the ciphertext c is invalid. Since $\mathcal{A}$ does not have access to E and $x([2\mathsf{m}_b + 1]P_E)$, then $x' = f_E(x([2\mathsf{m}_b + 1]P_E))$ is random, therefore the output b' of the query step is independent of b. Hence one expects to have roughly the same number on 1's and 0's after repeating the test T several times. This implies that

$$\Pr_T(1) = \frac{1}{2} \pm \mathsf{ngl}(\lambda).$$

We therefore construct our CSSIDDH solver $\mathcal{A}'$ as follows: if $\Pr_T(1) = \frac{1}{2} \pm \mathsf{ngl}(\lambda)$, then $\mathcal{A}'$ returns bad; if not, then $\mathcal{A}'$ returns $\mathsf{correct}$. □

Compared to the IND-CPA game setting, the adversary also has access to a decryption oracle $O(\cdot)$ in the IND-CCA game setting. To prove that InSIDH is IND-CCA secure, it is sufficient to prove that the decryption oracle is useless.

This immediately follows if we assume that no PPT adversary having access to E_0, E_1 and a valid ciphertext c, can produce a brand new valid ciphertext c' unless she encrypts c' herself. This is formalized in the following assumption.

Assumption 3 *The CSSIKoE (Commutative Supersingular Isogeny Knowledge of Exponent) assumption is stated as follows.*
Let λ be a security parameter, let $p = 2^r \ell_1 \cdots \ell_n - 1$ be a prime such that $\lambda + 2 \leq r \leq \frac{1}{2} \log p$. Let $[\mathfrak{a}]$ be a uniformly sampled element of $\mathsf{cl}(\mathbb{Z}[\pi])$. Let $\mathsf{c} = (E_3, x')$ be a valid ciphertext.
Then for every PPT adversary $\mathcal{A}$ that takes E_0, $[\mathfrak{a}]E_0$, c as inputs and returns a valid ciphertext $(F, y') \neq \mathsf{c}$, there exists a PPT adversary $\mathcal{A}'$ that takes the same inputs and returns $([\mathfrak{b}], F, y')$ where $[\mathfrak{b}] \in \mathsf{cl}(\mathbb{Z}[\pi])$ such that $F = [\mathfrak{b}]E_0$.

This assumption is analogous of the "knowledge of exponent" assumption (see Appendix A) introduced by Damgård in the context of discrete logarithm-based cryptography [11] and also used in [16].

Theorem 4. *Let us suppose that InSIDH is IND-CPA secure, and that Assumption 3 holds. Then InSIDH is IND-CCA secure.*

Proof. Let us suppose that Assumption 3 holds and InSIDH is not IND-CCA secure, and let us prove that InSIDH is not IND-CPA secure.

Since InSIDH is not IND-CCA secure, then there exists a PPT adversary $\mathcal{A}^{O(\cdot)} = (\mathcal{A}_1, O(\cdot))$ (where $O(\cdot)$ is the decryption oracle) that successfully determines if a given ciphertext c is that of a plaintext m_0 or m_1 with a non negligible advantage γ.

Suppose that the adversary $\mathcal{A}^{O(\cdot)}$ queries the decryption oracle $O(\cdot)$ with some valid ciphertexts $\mathsf{c}_1 = (F_1, x_1), \cdots, \mathsf{c}_n = (F_n, x_n)$ computed by $\mathcal{A}_1$. By Assumption 3, there exists a polynomial time algorithm $\mathcal{A}_2$ that when outputting $\mathsf{c}_1 = (F_1, x_1), \cdots, \mathsf{c}_n = (F_n, x_n)$ also outputs the ideal classes $[\mathfrak{b}_1], \cdots, [\mathfrak{b}_n]$ such that $F_i = [\mathfrak{b}_i]E_0$ for $i \in \{1, \cdots, n\}$. From the knowledge of the ideal classes $[\mathfrak{b}_1], \cdots, [\mathfrak{b}_n]$ and $[\mathfrak{a}]E_0$, the adversary $\mathcal{A}_2$ successfully decrypts $\mathsf{c}_1, \cdots, \mathsf{c}_n$.
Replacing the decryption oracle $O(\cdot)$ by $\mathcal{A}_2$, we obtain an adversary $\mathcal{A}' = (\mathcal{A}_1, \mathcal{A}_2)$ that successfully determines if a given ciphertext c is that of m_0 or m_1 with advantage γ (which is non negligible) and without making any call to the decryption oracle. This contradicts InSIDH's IND-CPA security. □

5 Implementation results

Here we present the experimentation results obtained by adapting the code of SiGamal [21]. The implementation is done using the two primes proposed by Moriya et al. for SiGamal.

SiGamal prime p_{128}. Let p_{128} be the prime $2^{130} \cdot \ell_1 \cdots \ell_{60} - 1$ where ℓ_1 through ℓ_{59} are the smallest distinct odd primes, and ℓ_{60} is 569. The bit length of p_{128} is 522. The private key bound is $m = 10$.

SiGamal prime p_{256}. Let p_{256} be the prime $2^{258} \cdot \ell_1 \cdots \ell_{43} - 1$ where ℓ_1 through ℓ_{42} are the smallest distinct odd primes, and ℓ_{43} is 307. The bit length of p_{256} is 515. The private key bound is $m = 32$.

All the costs (number of field multiplications, where $1\mathbf{S}{=}0.8\mathbf{M}$ and $1\mathbf{a}{=}0.05\mathbf{M}$) of CSIDH presented are done with the csidh-512 prime (of 512 bits) while those of InSIDH, SiGamal and C-SiGamal are with p_{128} and p_{256}. The costs presented in Table 1 and Table 2 are the average costs of $20,000$ rounds of key generation, encryption and decryption of each scheme.

Prime	csidh-512	p_{128}		p_{256}	
Scheme	CSIDH	InSIDH	(C)SiGamal	InSIDH	(C)SiGamal
Costs	$441,989$	$576,124$	$663,654$	$1,023,400$	$1,140,189$

Table 1: Cost of class group action for CSIDH, SiGamal, C-SiGamal and InSIDH.

	$p128$			$p256$		
	KGen	Enc.	Dec.	KGen	Enc.	Dec.
C-SiGamal	$663,594$	$1,433,805$	$767,176$	$1,151,447$	$2,685,714$	$1,528,020$
SiGamal		$1,326,856$	$760,861$		$2,208,530$	$1,536,829$
InSIDH	$576,124$	$1,159,533$	$679,733$	$1,023,827$	$2,057,297$	$1,417,401$

Table 2: Computational costs for C-SiGamal, SiGamal, and InSIDH.

Remark 2. In this proof of concept implementation, the class group algorithm considered does not take into account the improvements in [4], [2], [3].

6 Comparison with SiGamal and CSIDH

Here we compare InSIDH, (C-)SiGamal and CSIDH (or CSIDHpke more precisely). The comparison is done at four levels: design, security, keys and ciphertext sizes, and efficiency.

Design. At the design level, InSIDH seats between (C)SiGamal and CSIDH. InSIDH's private keys are ideal classes, as in CSIDH, while in (C)SiGamal they are integral ideals. In the class group action in (C-)SiGamal, a point has to be mapped through the isogeny as well, as opposed to CSIDH and InSIDH.

Securiy. Security wise, InSIDH IND-CPA security relies on CSIDH assumptions, contrarily to SiGamal whose IND-CPA security relies on new assumptions. Moreover, InSIDH is IND-CCA secure.

Keys and ciphertext sizes. The size of InSIDH's ciphertexts is equal to that of C-SiGamal's ciphertexts, and is half that of SiGamal ciphertexts. The size of InSIDH's public keys is half that of the public keys in SiGamal and C-SiGamal. The size of the private key in (C)SiGamal, compared to that of InSIDH, is augmented by r bits that are used to store the integer α such that the secret ideal $\mathfrak{a}$ is in the form $\mathfrak{a} = (\alpha)\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}$.

Efficiency. InSIDH is more efficient compared to SiGamal and C-SiGamal when using the same primes. From the results in Table 1, we have that for the prime p_{128}, the InSIDH class group action computation is 1.15x faster than that of

(C)SiGamal and is 1.30x slower than that of CSIDH; and for the prime prime p_{256}, it is 1.11x faster than that of (C)SiGamal and is 2.31x slower than that of CSIDH. For Encryption and decryption with the prime p_{128}, InSIDH is about 1.13x faster than SiGamal and about 1.19x faster than C-SiGamal. For the prime p_{256}, we get a 1.07x speedup when compared to SiGamal and a 1.21x speedup when compared to C-SiGamal.
We summarize the comparison in Table 3.

	CSIDHpke	InSIDH	SiGamal	C-SiGamal
Private key	$[\mathfrak{a}]$	$[\mathfrak{a}]$	$\mathfrak{a}$	$\mathfrak{a}$
Size of plaintext	$\log_2 p$	$r-2$	$r-2$	$r-2$
Size of Alice's public key	$\log_2 p$	$\log_2 p$	$2\log_2 p$	$2\log_2 p$
Size of ciphertexts (or Bob's public key)	$2\log_2 p$	$2\log_2 p$	$4\log_2 p$	$2\log_2 p$
Class group cost for p_{128} compared to CSIDH	x1.00	x1.30	x1.50	x1.50
Class group cost for p_{256} compared to CSIDH	x1.00	x2.31	x2.57	x2.57
Enc + Dec cost for p_{128} compared to CSIDHpke	x1.00	x1.38	x1.57	x1.65
Enc + Dec cost for p_{256} compared to CSIDHpke	x1.00	x2.62	x2.82	x3.17
Security	OW-CPA	IND-CCA	IND-CPA	IND-CPA

Table 3: Comparison between CSIDHpke, InSIDH, SiGamal and C-SiGamal.

7 Conclusion

In this paper, we revisited the protocols introduced by Moriya et al. at Asiacrypt2020, and obtained several results. We proved that the variant of SiGamal suggested by Moriya et al. is not IND-CCA secure. We construct a new isogeny based PKE scheme InSIDH by simplifying SiGamal in such a way that it resists the IND-CCA attack on SiGamal and its variants. InSIDH is more efficient than SiGamal and it has smaller private keys, public keys and ciphertexts. We prove that InSIDH is OW-CPA and IND-CPA secure relying on CSIDH assumptions. We introduce a Knowledge of Exponent assumption in the isogeny context. Relying on the later assumption, we prove that InSIDH is IND-CCA secure. Interestingly, InSIDH is also closer to CSIDH than SiGamal was, allowing for a better comparison between those two protocols.

The Knowledge of Exponent assumption we introduce is new, we leave a better study of it for future work. Nevertheless, it may have other applications in isogeny based cryptography.

Acknowledgements. We thank Tomoki Moriya, Hiroshi Onuki and Tsuyoshi Takagi for sharing the SiGamal magma code with us. We thank Ankan Pal for his help in running our magma code. We thank Tomoki Moriya and the anonymous reviewers for their useful feedback.

References

1. Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, Advances in Cryptology-EUROCRYPT'94, volume 950 of Lecture Notes in Computer Science, pages 92-111, Perugia,Italy, May 9-12, 1995. Springer, Heidelberg, Germany. (page 2)

2. Daniel J. Bernstein, Luca De Feo, Antonin Leroux, and Benjamin Smith. Faster computation of isogenies of large prime degree. Cryptology ePrint Archive, Report 2020/341, 2020. https://eprint.iacr.org/2020/341. (page 12)
3. Wouter Castryck and Thomas Decru. CSIDH on the surface. In J. Ding and J. P. Tillich, editors, *Post-quantum cryptography, 11th international conference, PQCrypto 2020*, volume 12100, pages 111–129. Springer, 2020. (page 4, 12)
4. Wouter Castryck, Thomas Decru, and Frederik Vercauteren. Radical Isogenies. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 493–519, Cham, 2020. Springer International Publishing. (page 12)
5. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 395–427. Springer, 2018. (page 2, 3, 4, 5)
6. Wouter Castryck, Jana Sotáková, and Frederik Vercauteren. Breaking the Decisional Diffie-Hellman Problem for Class Group Actions Using Genus Theory. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 92–120, Cham, 2020. Springer International Publishing. (page 5)
7. Denis X Charles, Kristin E Lauter, and Eyal Z Goren. Cryptographic hash functions from expander graphs. *Journal of Cryptology*, 22(1):93–113, 2009. (page 1)
8. Craig Costello and Huseyin Hisil. A simple and compact algorithm for SIDH with arbitrary degree isogenies. In: Takagi T., Peyrin T. (eds) Advances in Cryptology – ASIACRYPT 2017. ASIACRYPT 2017. Lecture Notes in Computer Science, vol 10625. Springer, Cham. https://doi.org/10.1007/978-3-319-70697-9_11. (page 3, 4)
9. Jean-Marc Couveignes. Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291, 2006. https://eprint.iacr.org/2006/291. (page 1)
10. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. Cryptology ePrint Archive, Report 1998/006, 1998. https://eprint.iacr.org/1998/006. (page 9)
11. Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 445–456, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg. (page 11)
12. Cyprien Delpech de Saint Guilhem, Péter Kutas, Christophe Petit, and Javier Silva. SÉTA: Supersingular encryption from torsion attacks. Cryptology ePrint Archive, Report 2019/1291, 2019. https://eprint.iacr.org/2019/1291. (page 2)
13. Christina Delfs and Steven D Galbraith. Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. *Designs, Codes and Cryptography*, 78(2):425–440, 2016. (page 3, 4)
14. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption. In Michael J. Wiener, editor, Advances in Cryptology-CRYPTO'99, volume 1666 of Lecture Notes in Computer Science, pages 537-554, Santa Barbara, CA, USA, August 15-19, 1999. Springer, Heidelberg, Germany. (page 2)
15. Steven D Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the security of supersingular isogeny cryptosystems. In *Advances in Cryptology – ASIACRYPT 2016*, pages 63–91. Springer, 2016. (page 9)
16. Satoshi Hada and Toshiaki Tanaka. On the existence of 3-round zero-knowledge protocols. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, pages 408–423, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. (page 11)

17. David Jao, Reza Azarderakhsh, Matthew Campagna, Craig Costello, Luca De Feo, Basil Hess, Aaron Hutchinson, Amir Jalali, Koray Karabina, Brian Koziel, Brian LaMacchia, Patrick Longa, Michael Naehrig, Geovandro Pereira, Joost Renes, Vladimir Soukharev, and David Urbanik. Supersingular Isogeny Key Encapsulation, October 1, 2020. `https://sike.org/files/SIDH-spec.pdf`. (page 1, 2)
18. David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, pages 19–34, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. (page 1)
19. Neal Koblitz. Elliptic curve cryptosystems. Math. Comp. 48 (1987), 203-209. (page 1)
20. E. Kummer. Zur theorie der complexen zahlen. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, pages 319 – 326, 1847. (page 4)
21. Tomoki Moriya. Magma codes for sigamal. Online, August 14, 2020. `http://tomoriya.work/code.html`. (page 2, 11)
22. Tomoki Moriya, Hiroshi Onuki, and Tsuyoshi Takagi. SiGamal: A Supersingular Isogeny-Based PKE and Its Application to a PRF. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2020*, pages 551–580, Cham, 2020. Springer International Publishing. (page 2, 7, 16)
23. Moni Naor. On cryptographic assumptions and challenges. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, pages 96–109, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg. (page 16)
24. National Institute of Standards and Technology: Post quantum Cryptography Standardization, December 2016. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization`. (page 1)
25. Stephen Pohlig and Martin Hellman. An improved algorithm for computing logarithms over gf(p) and its cryptographic significance. IEEE Transactions on information Theory, 24(1):106110, 1978. (page 6)
26. Joost Renes. Computing Isogenies Between Montgomery Curves Using the Action of (0, 0). In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography*, pages 229–247, Cham, 2018. Springer International Publishing. (page 3, 4)
27. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21 (2): 120–126, February 1978. `http://people.csail.mit.edu/rivest/Rsapaper.pdf`. (page 1)
28. Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer Science & Business Media, 2009. (page 3)
29. Anton Stolbunov. Constructing public-key cryptographic schemes based on class group action on a set of isogenous elliptic curves. *Adv. in Math. of Comm.*, 4(2):215–235, 2010. (page 1)
30. Lawrence C. Washington. *Elliptic Curves: Number Theory and Cryptography, Second Edition*. Chapman & Hall/CRC, 2 edition, 2008. (page 3)

A Knowledge of Exponent assumption

In the context of Discrete Logarithm-based cryptography, the Knowledge of Exponent assumption is stated as follows.

Assumption 4 (Knowledge of Exponent assumption [23]) *Let $G = \langle g \rangle$ be a cyclic group of prime order q where q is of cryptographic size. Let x be a uniformly random exponent in $\{2, \cdots, q-1\}$ and let $h = g^x$. The adversary tries to compute $h_1, h_2 \in G$ such that $h_1 = g^z$ and $h_2 = h^z$ for some $z \in \{2, \cdots, q-1\}$. The knowledge of exponent assumption holds if for every polynomial time adversary $\mathcal{A}$ that when given g, q and h outputs (g^z, h^z), there exists a polynomial time adversary $\mathcal{A}'$ that for the same inputs outputs (z, g^z, h^z).*

Intuitively, this assumption states that the only efficient way to compute (g^z, h^z) is to first fix z, then to compute g^z and h^z.

In InSIDH, the ciphertexts are of the form $\mathsf{c} = ([\mathfrak{b}]E_0, f_{[\mathfrak{b}][\mathfrak{a}]E_0}(x([2\mathsf{m}_0 + 1]P_{[\mathfrak{b}][\mathfrak{a}]E_0}))$. Assumption 3 states the only efficient way to compute a valid ciphertext is to first fix the ideal class $[\mathfrak{b}]$, then run the encryption algorithm of InSIDH to compute $\mathsf{c} = ([\mathfrak{b}]E_0, f_{[\mathfrak{b}][\mathfrak{a}]E_0}(x([2\mathsf{m}_0 + 1]P_{[\mathfrak{b}][\mathfrak{a}]E_0}))$.

B Generating the distinguished point of order 2^r

Here we discuss how when given a supersingular curve E defined over $\mathbb{F}_p$ where $p = 2^r \ell_1 \cdots \ell_n - 1$, one can efficiently generate a distinguished point P_E of order 2^r. The algorithm used by Moriya et al. in C-SiGamal to generate such a point mainly relies on the following result.

Theorem 5. *([22, Appendix A]) Let p be a prime such that $p \equiv 3 \bmod 4$ and let E be a supersingular Montgomery curve defined over $\mathbb{F}_p$ satisfying $End_{\mathbb{F}_p}(E) \cong \mathbb{Z}[\pi]$. Let $P \in E$.*
If $P \in E[\pi - 1] \setminus E[2]$, then $\quad x(P) \in (\mathbb{F}_p^*)^2 \iff P \in 2E[\pi - 1]$.
If $P \in E[\pi + 1] \setminus E[2]$, then $\quad x(P) \notin (\mathbb{F}_p^*)^2 \iff P \in 2E[\pi + 1]$.

Hence when searching for the x-coordinate of points of order 2^r in E, we need to avoid elements of $\mathbb{F}_p$ that are squares. Since $p = 2^r \ell_1 \cdots \ell_n - 1$ with $r > 1$, then $\left(\frac{-1}{p}\right) = -1$, $\left(\frac{2}{p}\right) = 1$ and $\left(\frac{\ell_i}{p}\right) = 1$ for $i \in \{1, \cdots, n\}$. Furthermore, let us suppose that $\ell_1, \cdots, \ell_{n-1}$ are the first smallest odd primes, then for every $I \subset \{0, 1, \cdots, n-1\}$, $\left(\frac{-\prod_{i \in I} \ell_i}{p}\right) = -1$ where $\ell_0 = 2$. Moriya et al.'s Algorithm [22, Appendix A] exploits this to consecutively sample x from the sequence $-2, -3, -4, \cdots$ and when x is the x-coordinate of a point in $E(\mathbb{F}_p)$, it checks if this point has order divisible by 2^r. Corollary 2 proves that if a such x is the x-coordinate of a point in $E(\mathbb{F}_p)$ then the corresponding point has order divisible by 2^r, hence the check is not necessary.

Corollary 2. *Let p be a prime such that $p \equiv 3 \bmod 4$ and let E be a supersingular Montgomery curve defined over $\mathbb{F}_p$ satisfying $End_{\mathbb{F}_p}(E) \cong \mathbb{Z}[\pi]$. Let $P \in E(\mathbb{F}_p)$ such that $x(P) \neq 0$.*
If $x(P) \notin (\mathbb{F}_p^*)^2$ then $[\ell_1 \times \cdots \times \ell_n]P$ is a point of order 2^r.

Proof. Since $E(\mathbb{F}_p) = E[\pi - 1]$ is a cyclic group, then there exist a point Q of order $p + 1 = 2^r \ell_1 \cdots \ell_n$ such that $E(\mathbb{F}_p) = \langle Q \rangle$. Set $P = [\alpha_P]Q$. Since

E is in the Montgomery form, then $E(\mathbb{F}_p) \cap E[2] = \langle (0,0) \rangle$. Since $x(P) \neq 0$, then $P \in E[\pi - 1] \setminus E[2]$. Let us suppose that $x(P) \notin (\mathbb{F}_p^*)^2$, then by Theorem 5 $P \notin 2E[\pi - 1]$, hence α_P is odd. Therefore, $\gcd(p+1, \alpha_P) = \gcd(2^r \ell_1 \cdots \ell_n, \alpha_P) = \gcd(\ell_1 \cdots \ell_n, \alpha_P)$. This implies that $P = [\alpha_P]Q$ is a point of order

$$\frac{p+1}{\gcd(p+1, \alpha_P)} = 2^r \cdot \frac{\ell_1 \cdots \ell_n}{\gcd(\ell_1 \cdots \ell_n, \alpha_P)}.$$

Hence $[\ell_1 \times \cdots \times \ell_n]P$ is a point of order 2^r.

Exploiting Corollary 2 we get Algorithm 1 which improves on that used by Moriya et al. for the same purpose.

Algorithm 1 Computing the distinguished point P_E

Require: The prime $p = 2^r \ell_1 \cdots \ell_n - 1$ and Montgomery coefficient $A \in \mathbb{F}_p$ of a supersingular curve.
Ensure: $P_E \in E(\mathbb{F}_p)$ of order 2^r.

```
Set x ← −2
while x^3 + Ax^2 + x is not a square in F_p and −x ≤ ℓ_{n−1} + 1 do
    Set x ← x − 1
if −x ≤ ℓ_{n−1} + 1 then
    Set P = (x, ·) ∈ E(F_p)
    Set P_E = [ℓ_1 × ··· × ℓ_n]P
    return P_E
else
    return ⊥.
```

A random element $x \in \mathbb{F}_p^* \setminus (\mathbb{F}_p^*)^2$ is the x-coordinate of a point $P \in E(\mathbb{F}_p)$ with probability $\frac{1}{2}$. The probability that Algorithm outputs $\perp$ is bounded by $\left(\frac{1}{2}\right)^{\ell_{n-1}}$. For SiGamal primes p_{256} and p_{128} (see Section 5), ℓ_{n-1} is 191 and 281 respectively, hence the output is $\perp$ with probability 2^{-191} and 2^{-281} respectively.

Remark 3. Algorithm 1 is deterministic, hence always outputs the same point P_E when the input in unchanged.