SimS: a Simplification of SiGamal

Tako Boris Fouotsa[1] and Christophe Petit[2,3]

[1] Università Degli Studi Roma Tre, Italy
takoboris.fouotsa@uniroma3.it
[2] Université Libre de Bruxelles, Belgium
[3] University of Birmingham's School of Computer Science, UK
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 SIDH and CSIDH, the new protocols provide IND-CPA security without the use of hash functions. 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 SimS, obtained by simplifying SiGamal. SimS has smaller public keys and ciphertexts than (C-)SiGamal and it is more efficient. We prove that SimS is IND-CCA secure under CSIDH security assumptions and one Knowledge of Exponent-type assumption we introduce. Interestingly, SimS is also much closer to the CSIDH protocol, facilitating a comparison between SiGamal and CSIDH.

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

1 Introduction

The construction of a large scale quantum computer would make the nowadays widely used public PKE schemes insecure, namely RSA [29], ECC [21] 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 [26].

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'. This hard problem was used by J. M. Couveignes [11], Rostovtsev and Stolbunov [30] to design a key exchange protocol using ordinary isogenies, and by Charles, Goren and Lauter [8] to design a cryptographic hash function using supersingular isogenies. In 2011, as a countermeasure to the sub-exponential quantum attack on the CRS (Couveignes-Rostovtsev-Stolbunov) scheme by Childs et

al. [9], Jao and De Feo designed SIDH [20] (Supersigular Isogeny Diffie-Hellman), a Key Exchange protocol based on supersingular isogenies. The submission of SIKE [19] (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 not the most efficient candidates for post quantum cryptography, but they provide the shortest keys and ciphertexts.

In 2018, Castryck et al. constructed CSIDH [6] (Commutative SIDH) using the $\mathbb{F}_p$-sub-graph of the supersingular isogeny graph. CSIDH key exchange is close to CRS but is an order of magnitude more efficient. PKE schemes based on isogeny problems include SIKE, SÉTA [14] and more recently SiGamal and C-SiGamal [24]. SÉTA and the PKEs canonically derived from the key exchange protocols SIDH and CSIDH are only OW-CPA secure. They require the use of hash functions and/or generic transformations such as the Fujisaki-Okamoto [16] or OAEP [1] to fulfil higher security requirements such as IND-CPA and IND-CCA security ([14, §2.4],[19, §1.4], [24, §3.3]). This motivated Moriya, Onuki and Tagaki to introduce the SiGamal [24, §5] and C-SiGamal [24, §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 [24], 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 which we call SimS. We prove that SimS 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 then introduce a "knowledge of Exponent" type assumption (Assumption 3) under which we prove that SimS is IND-CCA secure. This assumption may have other applications in isogeny-based cryptography.

We adapt the Magma code for SiGamal [23] to run a proof of concept implementation of SimS using the SiGamal primes p_{128} and p_{256}. For the prime p_{128}, SimS 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, SimS has smaller private keys, public keys and ciphertexts compared to SiGamal and C-SiGamal. SimS is simple, sits 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 1 best summarizes our contributions.

Outline. The remainder 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

	CSIDHpke	SimS	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 1: Comparison between CSIDHpke, SimS, SiGamal and C-SiGamal. CSIDHpke uses the csidh-512 prime, while SimS, SiGamal and C-SiGamal use the primes p_{128} and p_{256} which are SiGamal primes that provide the same security level as the csidh-512 prime.

action and the CSIDH key exchange protocol. In section 3, we present the SiGamal PKE scheme and we show that the variant suggested in [24, Remark 7] is not IND-CCA secure. Section 4 is devoted to SimS and its security arguments. In section 5 we present the outcome of a proof-of-concept implementation and compare SimS 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 $\mathcal{A}$,*

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

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

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

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 [31,32] for general mathematical background on supersingular elliptic curves and isogenies, to [6,15] for supersingular elliptic curves defined over $\mathbb{F}_p$ and their $\mathbb{F}_p$-endomorphism ring, and to [10,28] 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 curve E over $\mathbb{F}_p$. 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 $\#E(\mathbb{F}_p) \equiv 1 \mod p$, otherwise E is said to be ordinary. If E is a supersingular curve defined over $\mathbb{F}_p$ with $p > 3$, then $\#E(\mathbb{F}_p) = p + 1$. 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. In the rest rest of this section, we briefly describe the class group action used in CSIDH.

Let E be a supersingular curve defined over $\mathbb{F}_p$ and 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 $\mathbb{Z}[\frac{1+\pi}{2}]$ [15]. 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. *[6, Theorem 7] Let $\mathcal{O}$ 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 $\mathbb{F}_p$-isomorphism classes of curves in $\mathcal{E}\ell\ell_p(\mathbb{Z}[\pi])$ using the Montgomery coefficient A [4, Proposition 3].

The efficiency of the computation of an isogeny with known kernel essentially depends on the smoothness of its degree. In [6], 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 [22], $(\ell_i) = \mathfrak{l}_i\overline{\mathfrak{l}_i}$ 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 [10,28]. 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 [6, §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 [6] 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 [7], 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 SimS.

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

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. Recall that for every small odd prime ℓ_i dividing $p+1$, there are two prime ideals $\mathfrak{l}_i = (\ell_i, \pi - 1)$ and $\overline{\mathfrak{l}_i} = (\ell_i, \pi + 1)$ above ℓ_i in $\mathsf{cl}(\mathbb{Z}[\pi])$. Also, the isogenies $\phi_{\mathfrak{l}_i}$ and $\phi_{\overline{\mathfrak{l}_i}}$ of domain E_0 correspond to the isogenies with kernel generated by $P_{\mathfrak{l}_i} \in E_0[\ell_i] \cap ker(\pi - 1) \setminus \{0\}$ and $P_{\overline{\mathfrak{l}_i}} \in E_0[\ell_i] \cap ker(\pi + 1) \setminus \{0\}$ respectively. The points $\mathfrak{l}_i P_0$ and $\overline{\mathfrak{l}_i} P_0$ are images of the point P_0 trough these isogenies respectively. Let $\mathfrak{a} = (\alpha)\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n} \in \mathsf{cl}(\mathbb{Z}[\pi])$ where α is an integer then point $\mathfrak{a}P_0$ is the image of P_0 by the composition of the isogenies $\phi_{\mathfrak{l}_i}$ if $e_i > 0$ or $\phi_{\overline{\mathfrak{l}_i}}$ if $e_i < 0$, and the multiplication by α. For a given integer k, we denote by $[k] \circ \mathfrak{b}$ the composition of the isogeny corresponding to the ideal class $\mathfrak{b}$ and the scalar multiplication by k, and the point $[k] \circ \mathfrak{b}P_0$ denotes the image of P_0 trough this isogeny.

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 [27]. 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 [24] 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 [24, 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 [24, Remark 7] is not IND-CCA secure.*

4 SimS

We now introduce a new protocol that resists the previous attack. We name our protocol SimS (**Sim**plified **S**iGamal), which highlights the fact that our scheme is a simplification of 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 $\mathfrak{b}P_0$.

SimS is obtained by adjusting SiGamal in such a way that when a curve is part of the ciphertext, then none of its points are, and the other way around. In order to achieve this, we replace the point $\mathfrak{a}\mathfrak{b}P_0$ in the (C)SiGamal protocol by a canonical point $P_{E_4} \in E_4 = [\mathfrak{a}][\mathfrak{b}]E_0$. More concretely, in SimS, 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 $[\mathfrak{b}]$, 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

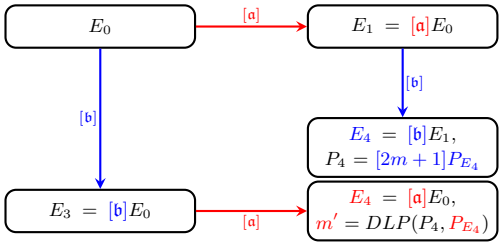

Fig. 1: SimS scheme. The elements in black are public, while those in blue are known only by Bob and those in red only by Alice.

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 depicts the scheme.

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

4.2 The SimS 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 its x-coordinate may leak too much information about the curve E_4. In fact $x(P_4)$ is the root of the 2^r division polynomial of E_4. Moreover, one could easily derive $x(P_4 + (0,0)) = \frac{1}{x(P_4)}$ by a simple inversion in $\mathbb{F}_p$, which would affect the IND-CCA security of the scheme. 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:

P1: f_E is bijective, f_E and its inverse $g_E = f_E^{-1}$ can be efficiently computed when E is given;
P2: for every element $x \in \mathbb{F}_p$, an adversary having no access to x and E cannot distinguish $f_E(x)$ from a random element of $\mathbb{F}_p$;
P3: 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 cannot 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 $\mathsf{bin}(x') = \mathsf{bin}(x) \oplus \mathsf{bin}(A_E)$ and $\mathsf{bin}(\cdot)$ takes an element in $\mathbb{F}_p$ and returns its binary representation.

Clearly, f_E is an involution, hence f_E is bijective and satisfies (P1). Proving that f_E satisfies (P2) and (P3) is less straightforward. Nevertheless, we give some intuitive arguments on why we believe that f_E satisfies (P2) and (P3). Given an element $y \in \mathbb{F}_p$, in order to distinguish whether $y = f_E(x)$ where x is the

x-coordinate of a point of order 2^r on some supersingular curve E, to the best of our knowledge, one needs to first fix the curve E, then check if $\mathsf{bin}(y) \oplus \mathsf{bin}(A_E)$ is the bit representation of the x-coordinate of a point of order order 2^r on E. This process needs to be repeated for all $O(\sqrt{p})$ supersingular elliptic curves defined over $\mathbb{F}_p$. Hence leading to an exponential adversary. The third property (P3), intuitively, follows from the fact there is no compatibility with XOR and algebraic operations. In fact, given $a \oplus b$, it seems hard to derive $R(a) \oplus b$ where R is non identical rational function.

Having such a function, SimS 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. *If f_{E_4} satisfies (P1), then SimS is correct.*

Proof. Since f_{E_4} satisfies (P1), then f_{E_4} is bijective, f_{E_4} and its inverse $g_{E_4} = f_{E_4}^{-1}$ can be efficiently computed by Alice since she has access to E_4.
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 SimS with SIDH would lead to a PKE scheme which is not IND-CCA secure because SIDH is vulnerable to adaptive attacks [17].

4.3 Security arguments

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

Theorem 3. *If Assumption 2 holds and f_{E_4} satisfies (P2), then SimS is IND-CPA secure.*

Proof. We adapt the proof of [12, Theorem 1] to our setting. Let us suppose that SimS 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 a SimS instance using two plaintext messages $\mathsf{m}_0, \mathsf{m}_1$ chosen by the adversary $\mathcal{A}$ and $(E_0, [\mathfrak{a}]E_0, [\mathfrak{b}]E_0, E)$. 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 $([\mathfrak{a}]E_0, \mathsf{c})$ 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 $E \neq [\mathfrak{a}][\mathfrak{b}]E_0$ and the tuple is bad. If in the query step $\mathcal{A}$ returns a bit b', then c is a valid ciphertext. Hence $[\mathfrak{a}][\mathfrak{b}]E_0 = E$, and the tuple is $\mathsf{correct}$.

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 $x([2\mathsf{m}_b + 1]P_E)$, and that f_E satisfies (P2), then $\mathcal{A}$ can not distinguish $x' = f_E(x([2\mathsf{m}_b + 1]P_E))$ from a random element of $\mathbb{F}_p$. 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 SimS 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 $[\mathfrak{a}]$, $[\mathfrak{b}]$ be a uniformly sampled elements of $\mathsf{cl}(\mathbb{Z}[\pi])$. Let $(f_E)_{E \in \mathsf{cl}(\mathbb{Z}[\pi])}$ be a family randomizing functions as defined in Section 4.2 such that each of these functions satisfies (P3).

Then for every PPT adversary $\mathcal{A}$ that takes E_0, $[\mathfrak{a}]E_0$ and $([\mathfrak{b}]E_0, f_{[\mathfrak{a}][\mathfrak{b}]E_0}(x(P)))$ where $P \in [\mathfrak{a}][\mathfrak{b}]E_0$ is a point of order 2^r as inputs, and returns a couple $([\mathfrak{b}']E_0, f_{[\mathfrak{a}][\mathfrak{b}']E_0}(x(P'))) \neq ([\mathfrak{b}]E_0, f_{[\mathfrak{a}][\mathfrak{b}]E_0}(x(P)))$ where $P' \in [\mathfrak{a}][\mathfrak{b}']E_0$ is a point of order 2^r, there exists a PPT adversary $\mathcal{A}'$ that takes the same inputs and returns $([\mathfrak{b}'], [\mathfrak{b}']E_0, f_{[\mathfrak{a}][\mathfrak{b}']E_0}(x(P'))))$.

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

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

Since SimS 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 SimS's IND-CPA security. □

Remark 2. In all this section, we have assumed that the ideal classes $[\mathfrak{a}]$ and $[\mathfrak{b}]$ were uniformly sampled elements of $\mathsf{cl}(\mathbb{Z}[\pi])$. Strictly speaking, in order to uniformly sample elements in $\mathsf{cl}(\mathbb{Z}[\pi])$, one needs to compute the class group structure and its generators. Computing the class group $\mathsf{cl}(\mathbb{Z}[\pi])$ requires sub-exponential time in its discriminant [3, §1]. The class group structure for the CSIDH-512 prime was computed in [3] with a lot of computational effort. As in the preliminary version of CSIDH or instantiations of CSIDH using different primes for which the class group is unknown, we assume that the many small prime ideals $\mathfrak{l}_i$ used to sampled elements in $\mathsf{cl}(\mathbb{Z}[\pi])$ (see Section 2.2) generate the entire class group or a sufficiently large subgroup of the class group such that the sampled ideals are close to being uniformly random. See [6, §7.1] for more details.

Remark 3. The secret vectors $(e_1, \cdots, e_n) \in [-m, m]^n$ used to sample ideals $\mathfrak{a} = \mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n} \in \mathsf{cl}(\mathbb{Z}[\pi])$ can be seen as analogous to exponents in discrete logarithm-based protocols, and Assumption 3 is in that sense analogous to the "knowledge of exponent" assumption (see Appendix A) introduced by Damgård in the context of discrete logarithm-based cryptography [13] and also used in [18]. If ever the class group $\mathsf{cl}(\mathbb{Z}[\pi])$ were computed for the SimS primes, then the analogy would be more immediate.

5 Implementation results

Here we present the experimentation results obtained by adapting the code of SiGamal [23]. 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 SimS, SiGamal and C-SiGamal are with p_{128} and p_{256}. The costs presented in Table 2 and Table 3 are the average costs of $20,000$ rounds of key generation, encryption and decryption of each scheme.

Remark 4. In this proof of concept implementation, the class group algorithm considered does not take into account the improvements in [5], [2], [4].

6 Comparison with SiGamal and CSIDH

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

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

Table 2: Cost (number of field multiplications, where $1\mathbf{S}=0.8\mathbf{M}$ and $1\mathbf{a}=0.05\mathbf{M}$) of class group action for CSIDH with the csidh-512 prime, SimS, SiGamal and C-SiGamal with p_{128} and p_{256}.

	p_{128}			p_{256}		
	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		$1,326,856$	$760,861$		$2,208,530$	$1,536,829$
SimS	$576,124$	$1,159,533$	$679,733$	$1,023,827$	$2,057,297$	$1,417,401$

Table 3: Computational costs (number of field multiplications, where $1\mathbf{S}=0.8\mathbf{M}$ and $1\mathbf{a}=0.05\mathbf{M}$) for C-SiGamal, SiGamal and SimS with p_{128} and p_{256}.

Design. At the design level, SimS sits between (C)SiGamal and CSIDH. SimS'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 SimS.

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

Keys and ciphertext sizes. The size of SimS's ciphertexts is equal to that of C-SiGamal's ciphertexts, and is half that of SiGamal ciphertexts. The size of SimS'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 SimS, 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. SimS is more efficient compared to SiGamal and C-SiGamal when using the same primes. From the results in Table 2, we have that for the prime p_{128}, the SimS 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}, SimS 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 1. Note that the encryption in CSIDH-pke is essentially two CSIDH class group computations and the decryption is one class group computation.

7 Conclusion

In this paper, we revisited the protocols introduced by Moriya et al. at Asiacrypt 2020, 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 SimS by simplifying SiGamal in such a way that it resists the IND-CCA attack on SiGamal and its variants. SimS is more efficient than SiGamal and it has smaller private keys, public keys and ciphertexts. We prove that SimS is 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 SimS is IND-CCA secure. Interestingly, SimS is also closer to CSIDH than SiGamal was, allowing for a better comparison between those two protocols.

We leave a better study of the Knowledge of Exponent assumption and further cryptographic applications of this assumption to future work.

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. Bellare, M., Rogaway, P.: 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
2. Bernstein, D.J., Feo, L.D., Leroux, A., Smith, B.: Faster computation of isogenies of large prime degree. Cryptology ePrint Archive, Report 2020/341 (2020), https://eprint.iacr.org/2020/341
3. Beullens, W., Kleinjung, T., Vercauteren, F.: CSI-FiSh: Efficient Isogeny Based Signatures Through Class Group Computations. In: Galbraith, S.D., Moriai, S. (eds.) Advances in Cryptology – ASIACRYPT 2019. pp. 227–247. Springer International Publishing, Cham (2019)
4. Castryck, W., Decru, T.: CSIDH on the surface. In: Ding, J., Tillich, J.P. (eds.) Post-quantum cryptography, 11th international conference, PQCrypto 2020. vol. 12100, pp. 111–129. Springer (2020), http://dx.doi.org/10.1007/978-3-030-44223-1_7
5. Castryck, W., Decru, T., Vercauteren, F.: Radical Isogenies. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 493–519. Springer International Publishing, Cham (2020)
6. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: CSIDH: an efficient post-quantum commutative group action. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 395–427. Springer (2018)
7. Castryck, W., Sotáková, J., Vercauteren, F.: Breaking the Decisional Diffie-Hellman Problem for Class Group Actions Using Genus Theory. In: Micciancio,

D., Ristenpart, T. (eds.) Advances in Cryptology – CRYPTO 2020. pp. 92–120. Springer International Publishing, Cham (2020)
8. Charles, D.X., Lauter, K.E., Goren, E.Z.: Cryptographic hash functions from expander graphs. Journal of Cryptology **22**(1), 93–113 (2009)
9. Childs, A., Jao, D., Soukharev, V.: Constructing elliptic curve isogenies in quantum subexponential time. Journal of Mathematical Cryptology **8**(1), 1–29 (2014)
10. Costello, C., Hisil, H.: 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
11. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive, Report 2006/291 (2006), https://eprint.iacr.org/2006/291
12. Cramer, R., Shoup, V.: 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
13. Damgård, I.: Towards practical public key systems secure against chosen ciphertext attacks. In: Feigenbaum, J. (ed.) Advances in Cryptology — CRYPTO '91. pp. 445–456. Springer Berlin Heidelberg, Berlin, Heidelberg (1992)
14. De Saint Guilhem, C.D., Kutas, P., Petit, C., Silva, J.: SÉTA: Supersingular encryption from torsion attacks. Cryptology ePrint Archive, Report 2019/1291 (2019), https://eprint.iacr.org/2019/1291
15. Delfs, C., Galbraith, S.D.: Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$. Designs, Codes and Cryptography **78**(2), 425–440 (2016)
16. Fujisaki, E., Okamoto, T.: 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
17. Galbraith, S.D., Petit, C., Shani, B., Ti, Y.B.: On the security of supersingular isogeny cryptosystems. In: Advances in Cryptology – ASIACRYPT 2016. pp. 63–91. Springer (2016)
18. Hada, S., Tanaka, T.: On the existence of 3-round zero-knowledge protocols. In: Krawczyk, H. (ed.) Advances in Cryptology — CRYPTO '98. pp. 408–423. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
19. Jao, D., Azarderakhsh, R., Campagna, M., Costello, C., Feo, L.D., Hess, B., Hutchinson, A., Jalali, A., Karabina, K., Koziel, B., LaMacchia, B., Longa, P., Naehrig, M., Pereira, G., Renes, J., Soukharev, V., Urbanik, D.: Supersingular Isogeny Key Encapsulation (October 1, 2020), https://sike.org/files/SIDH-spec.pdf
20. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B.Y. (ed.) Post-Quantum Cryptography. pp. 19–34. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
21. Koblitz, N.: Elliptic curve cryptosystems. Math. Comp. 48 (1987), 203-209
22. Kummer, E.: Zur theorie der complexen zahlen. Journal für die reine und angewandte Mathematik (Crelles Journal) pp. 319 – 326 (1847)
23. Moriya, T.: Magma codes for sigamal. Online (August 14, 2020), http://tomoriya.work/code.html
24. Moriya, T., Onuki, H., Takagi, T.: SiGamal: A Supersingular Isogeny-Based PKE and Its Application to a PRF. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 551–580. Springer International Publishing, Cham (2020)

25. Naor, M.: On cryptographic assumptions and challenges. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003. pp. 96–109. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
26. National Institute of Standards and Technology: Post-quantum Cryptography Standardization (December 2016), https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization
27. Pohlig, S., Hellman, M.: An improved algorithm for computing logarithms over gf(p) and its cryptographic significance. IEEE Transactions on information Theory, 24(1):106110, 1978
28. Renes, J.: Computing Isogenies Between Montgomery Curves Using the Action of (0, 0). In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography. pp. 229–247. Springer International Publishing, Cham (2018)
29. Rivest, R., Shamir, A., Adleman, L.: 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
30. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. IACR Cryptol. ePrint Arch. **2006**, 145 (2006)
31. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer Science & Business Media (2009)
32. Washington, L.C.: Elliptic Curves: Number Theory and Cryptography, Second Edition. Chapman & Hall/CRC, 2 edn. (2008)

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 [25]) *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 SimS, 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 SimS 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. *([24, 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 [24, 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.

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

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 $x \leftarrow -2$
while $x^3 + Ax^2 + x$ is not a square in $\mathbb{F}_p$ and $-x \leq \ell_{n-1} + 1$ **do**
    Set $x \leftarrow x - 1$
if $-x \leq \ell_{n-1} + 1$ **then**
    Set $P = (x, \cdot) \in E(\mathbb{F}_p)$
    Set $P_E = [\ell_1 \times \cdots \times \ell_n]P$
    return P_E
else
    return $\perp$.