A Post-Quantum Signcryption Scheme Using Isogeny Based Cryptography

Kunal Dey[1], Sumit Kumar Debnath[2,3*], Pantelimon Stănică[4] and Vikas Srivastava[5]

[1]Department of Mathematics, National Institute of Technology Jamshedpur, Jamshedpur-831014, India; kunaldey3@gmail.com
[2]Department of Mathematics, National Institute of Technology Jamshedpur, Jamshedpur-831014, India; sd.iitkgp@gmail.com
[3]Department of Mathematics, Indian Institute of Information Technology Kalyani, Kalyani – 741235, India
[4]Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA 93943, USA; pstanica@nps.edu
[5]Department of Mathematics, National Institute of Technology Jamshedpur, Jamshedpur-831014, India; vikas.math123@gmail.com

Abstract

Signcryption is an important cryptographic scheme which is used for both confidentiality and unforgeability. It has many interesting practical applications. Enormous growth of quantum computers makes a warning to the existing classical signcryption schemes due to Shor's algorithm. As a result, designing signcryption schemes, which can withstand quantum attack, is an interesting direction of research. Isogeny based cryptography (IBC) is an ideal post-quantum candidate that can be employed to build a quantum computer immune signcryption scheme. Less communication cost and a smaller public key is the main advantage of IBC compared to other post quantum cryptographic branches. In this paper, we design the *first* sgncryption employing IBC. Our scheme is relying on three hard problems: Commutative Supersingular Isogeny Decisional Diffie-Hellman (CSSIDDH), Group Action Inverse Problem (GAIP) and Commutative Supersingular Isogeny Knowledge of Exponent (CSSIKOE). It achieves $\mathsf{IND}-\mathsf{CCA}$ and $\mathsf{EUF}-\mathsf{CMA}$ security. Ciphertext size in this scheme turns out to be 16622.05 bytes for p_{128} and 12757.45 bytes for p_{256} to achieve NIST-1 level of security.

Keywords: isogeny based cryptography; post-quantum cryptography; digital signature; public key encryption; signcryption.

MSC 2020: 94A60; 68M12; 68P25; 68P30.

1 Introduction

Public key encryption (PKE) and digital signature are two widely used important cryptographic primitives. PKE enables confidentiality, whereas digital signature allows authentication, integrity and non-repudiation. In many practical scenarios, we need authentication, integrity, non-repudiation and confidentiality simultaneously. For example, establishment secure and authenticated communication system like email should attain those features simultaneously. As a consequence, signcryption arrives into the picture to enable a party to simultaneously perform

the functions of both digital signature and encryption. It is run between a sender and a receiver. The following properties must be satisfied by a valid signcryption scheme:

Correctness: On receiving a valid ciphertext of a message m from the sender, the receiver should be able to decrypt it and verify the message m efficiently.

Efficiency: The computation cost of a signcryption scheme should be less than the sum of the computation costs of an underlying encryption and a signature scheme. Similarly, the communication cost of a signcryption scheme should be smaller than the sum of the communication costs of an underlying encryption and a signature scheme.

Confidentiality: Without knowing the secret keys of the sender and the receiver, it is infeasible for an attacker to gain any knowledge about the plaintext message m.

Unforgeability: It should be infeasible to simulate an honest sender by an attacker and produce a valid ciphertext.

Integrity: The receiver should be able to verify that the received message is the same as the message that has been signcrypted by the sender.

Non-repudiation: When required, the receiver should be able to prove to some third party that the ciphertext is generated by the sender.

In 1997, the first ever signcryption scheme was proposed by Zheng [16]. Since then, several signcryption schemes [8, 11, 15] have been designed. Most of them are relying on hard number theoretical problems such as "discrete logarithm problem", "integer factorization problem", etc. The expected growth of quantum computers will become a huge security threat since Shor's algorithm [10] can solve efficiently some of those problems in polynomial time, or at the least, impact their security, using sufficient number of quantum computers. As a result, post quantum cryptography (PQC) has come in this context to provide security against a quantum computer. Lattice based cryptography, multivariate cryptography, isogeny based cryptography (IBC, hash based cryptography, code based cryptography are the main proposals for PQC. Among these, IBC, which was initiated by Couveignes [2] in 1997, has received considerable attention due to its salient feature of low communication bandwidth. In the last few decades, several works [9, 25, 28] have been done in the context of IBC. However, there are only few constructions of IBC based digital signatures. Galbraith et al. [27] came up with a signature scheme based on quaternions of l-isogeny problem. In the following, the signature SeaSign was proposed by Feo et al. [4]. They employed CSIDH [1] as the cryptographic building block. Beullens et al. [26] developed CSI-FIsh with the help of CSIDH [1]. Recently, Feo et al. [25] presented a digital signature scheme SQISign by using quaternions and isogeny. In the current state of art, a variety of signcryption schemes [6, 7, 12, 14] have been proposed in the context of PQC. However, there is no construction of signcryption scheme based on IBC. Thus, developing a signcryption utilizing IBC remains an interesting direction of further research.

1.1 Application

- Some digital communication requires confidentiality and authenticity. One of the popular electronic communication protocol is Secure Electronic Transaction (SET) [17]. Signcryption is used to secure this kind of protocol.

- In an e-cash system [18, 19], clients want to spend their electronic cash in incognito mode. Here blind signcryption scheme can be implemented to design such scheme.
- Apart from important application of signcryption includes Secure/Multipurpose Internet Mail Extensions (S/MIME), Hypertext Transfer Protocol Secure (HTTPS), network communication protocol (SSH).

1.2 Our Contribution

Due to Shor's algorithm [10], one can break the security of public key cryptography (PKC) based upon some number theoretical problems (like integer factorization) with sufficiently large quantum computer, in polynomial time. Even if the PKC is not based upon factorization, a quantum computer may still dent its security. As a consequence, signcryption based on PKC will face a serious security threat, once the large scale quantum computers are built. Thus, we need an alternative solution that can withstand future quantum attacks. The employment of IBC in the context of signcryption yields post-quantum signcryption that remains immune against a quantum attack. The low communication cost of IBC makes it attractive as compared to the other PQC candidates. In this paper, we deal with constructions and analysis of a signcryption scheme relying on IBC. In particular, our signcryption scheme is the first that employs IBC to achieve post-quantum security. The signature scheme of [4] and the encryption scheme of [5] have been utilized as the fundamental primitives of our design. Both of the above schemes are rely on Commutative Supersingular Isogeny Diffie-Hellman (CSIDH) [1]. $\mathsf{IND-CCA}$ and $\mathsf{EUF-CMA}$ security of our proposed scheme are guaranteed by the believed hardness of the following problems: Commutative Supersingular Isogeny Decisional Diffie-Hellman ($\mathsf{CSSIDDH}$), Group Action Inverse Problem (GAIP) and Commutative Supersingular Isogeny Knowledge of Exponent ($\mathsf{CSSIKOE}$). To achieve NIST-1 level of security, we show that the size of the ciphertext for the proposed scheme remains 16622.05 bytes for p_{128} and 12757.45 bytes for p_{256} which is less than the sum of individual ciphertext size and signature size, generated by the encryption scheme and the signature scheme, respectively. We also calculate the same parameters for other CSIDH variants.

2 Preliminaries

Notation:
$a \in_R A$ stands for selecting an element a from a set A uniformly at random. Further, $\bar{\mathbb{F}}$ represents the algebraic closure of the field $\mathbb{F}$ and $\mathbb{F}^* = \mathbb{F} - \{0\}$, $\#X$ stands for the cardinality of a set X and π_E denotes the Frobenius endomorphism of an elliptic curve E over some finite field.

Definition 2.1. ***Negligible function:***
A negligible function negl *is a function from the set of natural number* ($\mathbb{N}$) *to the set of real number* ($\mathbb{R}$) *so that for every polynomial* $poly(x)$ *in* x*, there exists a positive integer* N *such that* $\mathsf{negl}(x) < \frac{1}{poly(x)}$, $\forall\, x > N$.

2.1 Elliptic Curve [3]

For any field $\mathbb{F}$, the projective space of dimension n is denoted by $\mathcal{P}^n$ which consists of the equivalence classes of all points $(y_1, y_2, \ldots, y_{n+1}) \neq (0, 0, \ldots, 0)$ in $\bar{\mathbb{F}}^{n+1}$ via the equivalence relation $(y_1, y_2, \ldots, y_{n+1}) \sim (z_1, z_2, \ldots, z_{n+1})$ such that $y_i = \lambda z_i$ for some $\lambda \in \mathbb{F}^*$. Any point in the projective space can be written as $(y_1 : y_2 : \ldots : y_{n+1})$. In the projective space $\mathcal{P}^2$, the Weierstrass form of an elliptic curve E is denoted by $Y^2Z = X^3 + cXZ^2 + dZ^3$ with $c, d \in \mathbb{F}$ and discriminant $4c^3 + 27d^2 \neq 0$. We let $\Theta = (0 : 1 : 0)$ be the point of infinity of the curve (this is the identity element for the group structure, which an elliptic curve is endowed with). We can transform the above curve to affine coordinates by substituting $x = \frac{X}{Z}$, $y = \frac{Y}{Z}$, and so the reduced equation can therefore be written as $y^2 = x^3 + cx + d$, $c, d \in \mathbb{F}$. Any projective curve of genus 1 over a field $\mathbb{F}$ with $char(F) \neq 2, 3$, is isomorphic to the aforementioned Weierstrass form of an elliptic curve. The $\mathbb{F}$-rational points of an elliptic curve over $\mathbb{F}$ form a group which is denoted by $E(\mathbb{F})$. If $\mathbb{F}$ is a number field then $E(\mathbb{F})$ is finitely generated. The m-torsion group $E[m]$ is the set of all points $P \in E(\bar{\mathbb{F}})$ such that $mP = \Theta$. For $char(\mathbb{F}) = p$, E is supersingular if $E[p] \cong \Theta$, whereas E is ordinary if $E[p] \cong \frac{\mathbb{Z}}{p\mathbb{Z}}$.

A surjective morphism between two elliptic curve E and E' is called an isogeny between E and E'. Indeed, this may not be injective, i.e, it may have nontrivial kernel. If $E = E'$, we call an isogeny an endomorphism and the collection of all endomorphisms on E along with the zero map is called the endomorphism ring, denoted by $End(E)$. Multiplication-by-m map, $[m] : P \longrightarrow [m]P$, $P \in E(\bar{\mathbb{F}})$ is a fundamental example of an endomorphism. For a finite field $\mathbb{F}_q$, the Frobenius endomorphism of E over $\mathbb{F}_q$ is denoted by $\pi_E : (X : Y : Z) \longleftrightarrow (X^q : Y^q : Z^q)$. Endomorphisms which are defined over $\mathbb{F}_p$, are called $\mathbb{F}_p$-rational endomorphisms, whose set is denoted by $End_p(E)$.

2.2 Class Group Action [3, 9]

The endomorphism ring $End(E)$ is a $\mathbb{Z}$-algebra. We can transform that $\mathbb{Z}$-algebra into a $\mathbb{Q}$-algebra by taking the tensor product of $End(E)$ with $\mathbb{Q}$. Such a $\mathbb{Q}$-algebra is called an endomorphism algebra and denoted by $End^0(E)$, which indeed is a $\mathbb{Q}$-vector space. The dimension of $End^0(E)$ can be 1, 2 or 4 depending on whether $End^0(E)$ is isomorphic to $\mathbb{Q}$, to an imaginary quadratic field or to a quaternion algebra. An order $\mathscr{O}$ of a $\mathbb{Q}$-algebra L is a free $\mathbb{Z}$-module. The rank of $\mathscr{O}$ is the dimension of L as a $\mathbb{Q}$-vector space. Suppose M is a number field of dimension r as a $\mathbb{Q}$-vector space. Then the set of algebraic integers $\mathscr{O}_M$ forms an order of rank r, which is the unique maximal order of M. With an order $\mathscr{O}$ and an $\mathscr{O}$-ideal $\mathfrak{a}$, we can define a fractional $\mathscr{O}$-ideal as $\mathfrak{b} = \omega\mathfrak{a} = \{\omega\alpha : \alpha \in \mathfrak{a}\}$ for some $\omega \in M^*$. If the fractional $\mathscr{O}$-ideal $\mathfrak{b}$ lies in $\mathscr{O}$ then $\mathfrak{b}$ will be an $\mathscr{O}$-ideal. On the other hand, every $\mathscr{O}$-ideal is a fractional $\mathscr{O}$-ideal. A fractional $\mathscr{O}$-ideal $\mathfrak{b}$ is called invertible if there exists a fractional $\mathscr{O}$-ideal $\mathfrak{c}$ such that $\mathfrak{bc} = \mathscr{O}$. The collection of all invertible fractional $\mathscr{O}$-ideals forms a group with respect to the multiplication of fractional $\mathscr{O}$-ideals which is defined as $\mathfrak{bb}' = \omega\omega'\mathfrak{aa}'$ for two fractional $\mathscr{O}$-ideals $\mathfrak{b} = \omega\mathfrak{a}$ and $\mathfrak{b}' = \omega'\mathfrak{a}'$. Let the group is denoted by I and its subgroup consisting of principal fractional $\mathscr{O}$-ideals is denoted by P. The group $cl(\mathscr{O}) = \frac{I}{P}$ denotes the ideal class group. This group is a finite commutative group and its cardinality is known as class number.

For a supersingular elliptic curve E over $\mathbb{F}_p$, $\mathscr{O} = End_p(E)$ is an order in an imaginary quadratic field. We denote $Ell_p(\mathscr{O}) = \{E : E \; is \; defined \; over \; \mathbb{F}_p \; and \; End_p(E) = \mathscr{O}\}$. The action of $cl(\mathscr{O})$ on $Ell_p(\mathscr{O})$ can be defined as $\mathfrak{a}E_{\mathfrak{b}} = E_{\mathfrak{a}^{-1}\mathfrak{b}}$, where $\mathfrak{ab} \in cl(\mathscr{O})$.

Suppose an elliptic curve E is defined over $\mathbb{F}_p$ with $p \geq 5$. The Frobenius endomorphism π_E satisfies $\pi_E^2 - t\pi_E + p = 0$, where $t = trace(\pi_E)$. If E is supersingular then $t = 0$ which implies $\pi^2 + p = 0$. Since $\pi_E \in End_p(E)$ and $End_p(E)$ is an order of $\mathbb{Q}(\sqrt{-p})$, one can write $End_p(E) = \mathbb{Z}[\pi_E]$, where $\mathbb{Z}[\pi_E]$ is the subring of $\mathbb{Q}(\sqrt{-p})$ generated by π_E.

The following theorems give conditions for which elliptic curve is isomorphic to Montgomery curve. The reader can see [3, 9] for more details.

Theorem 2.2. *[1] Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$, where $p \equiv 3 \pmod 8$. The necessary and sufficient condition for $End_p(E) = \mathbb{Z}[\pi_E]$ (π_E is the Frobenius map) is that there exists a unique $A \in \mathbb{F}_p$ so that E is isomorphic to a Montgomery curve: $y^2 = x^3 + Ax^2 + x$.*

Theorem 2.3. *[1] Let E be a supersingular elliptic curve defined over $\mathbb{F}_p$, where $p \equiv 3 \pmod 4$ and $p > 3$. The necessary and sufficient condition for $End_p(E) = \mathbb{Z}[\pi_E]$ (π_E is the Frobenius map) is that there exists a unique $A \in \mathbb{F}_p$ so that E is $\mathbb{F}_p$ isomorphic to a Montgomery curve: $y^2 = x^3 + Ax^2 + x$.*

2.3 Commutative Supersingular Isogeny Diffie-Hellman (CSIDH) [1]

CSIDH is a key-agreement protocol that allows two party Alice and Bob to share the same key. In CSIDH [1], an elliptic curve $E : y^2 = x^3 + x$ over $\mathbb{F}_p$ is taken for a prime of the form $p = 4k_1k_2 \dots k_n - 1$ where $k_1, k_2, \dots, k_n$ are small distinct odd primes. As $p \equiv -1 (\text{mod } 4)$, one can say that E is supersingular. Hence, $|E(\mathbb{F}_p)| = p + 1 \equiv 0 (\text{mod } k_i)$ and one can decompose $k_i\mathscr{O} = J_i\overline{J_i}$, where $J_i = (k_i, \pi_E - 1)$ and $\overline{J_i} = (k_i, \pi_E + 1)$. Indeed, $\mathscr{O} = Z[\pi_E] = End_p(E)$ and $|cl(Z[\pi_E])| = \#\{[J_1]^{a_1}[J_2]^{a_2} \dots [J_n]^{a_n} : a_1, a_2, \dots, a_n \in \{-k, \dots, k\}\}$ for some integer k. A classic ideal can be represented as $(a_1, a_2, \dots, a_k)$. By Theorem 2.3, E is $\mathbb{F}_p$-isomorphic to a Montgomery curve $E_A : y^2 = x^3 + Ax^2 + x$, which is represented by its coefficient A.

For generating a public key/private key pair $(A, [\mathbf{a}])$, Alice randomly chooses $\mathbf{a} = (a_1, a_2, \dots, a_k)$ from $\{-k, \dots, k\}$, computes $[\mathbf{a}] = [J_1^{a_1} J_2^{a_2} \cdots J_k^{a_k}]$ and evaluates the group action $[\mathbf{a}]E$ which is isomorphic to a Montgomery curve E_A with coefficient A. Similarly, Bob computes his key pair as $(B, [\mathbf{b}])$ corresponding to a Montgomery curve E_B with coefficient B. On receiving the public key from the other, Alice evaluates $[\mathbf{a}]E_B = [\mathbf{a}][\mathbf{b}]E$ and Bob calculates $[\mathbf{b}]E_A = [\mathbf{b}][\mathbf{a}]E$. Due to the commutative property of $\mathbb{Q}(\sqrt{-p})$, they are able to share the same key $[\mathbf{a}][\mathbf{b}]E = [\mathbf{b}][\mathbf{a}]E$.

2.4 Randomize Function Indexed by a Supersingular Elliptic Curve [5]

For a supersingular elliptic curve E over $\mathbb{F}_p$ a function, $f_E : \mathbb{F}_p \longleftarrow \mathbb{F}_p$ is said to be a randomize function indexed by E if it satisfies the following properties:

`Property 1`: f_E is bijective and given E, one can access both f_E and f_E^{-1}.

`Property 2`: Without any access to $x \in \mathbb{F}_p$ and E, an adversary cannot distinguish between $f_E(x)$ and a random element of $\mathbb{F}_p$.

`Property 3`: For any rational function $F \in \mathbb{F}_p[x]$, an adversary cannot evaluate $f_E(F(x))$ without any access to $x \in \mathbb{F}_p$ and E.

2.5 Hardness Assumptions

`Hardness assumption 1:`

Definition 2.4. *Commutative Supersingular Isogeny Decisional Diffie-Hellman (*CSSIDDH*)* [9]
Suppose $k_1, k_2, \ldots, k_n$ are n distinct small odd primes such that $q = 2^m k_1 k_2 \ldots k_n - 1$. We choose a supersingular elliptic curve $E : y^2 = x^3 + x$ over $\mathbb{F}_q$ with the Frobenius endomorphism π_E. We say that the CSSIDDH *holds provided the*

$$\left| Pr\left[\delta = \delta' : \begin{array}{c} [\boldsymbol{a}], [\boldsymbol{b}], [\boldsymbol{c}] \in_R cl(\mathbb{Z}[\pi_E]), \\ K_0 = [\boldsymbol{a}][\boldsymbol{b}]E, K_1 = [\boldsymbol{c}]E, \\ \delta \in \{0,1\}, \\ \text{Given } (E, E, [\boldsymbol{b}]E, K_\delta), \text{ the algorithm outputs a bit } \delta' \end{array} \right] - \tfrac{1}{2} \right| < \mathsf{negl}(\kappa)$$

for any probabilistic polynomial time (PPT) algorithm $\mathcal{A}$.

`Hardness assumption 2:`

Definition 2.5. *Group Action Inverse Problem (*GAIP*)* [4]
Given two elliptic curves E and E' over $\mathbb{F}_q$ with $E' = [\boldsymbol{a}]E$ for $[\boldsymbol{a}] \in cl(\mathbb{Z}[\pi_E])$, it is hard to find $[\boldsymbol{a}]$.

`Hardness assumption 3:`

Definition 2.6. *Commutative Supersingular Isogeny Knowledge Of Exponent (*CSSIKOE [5]
Let $k_1, k_2, \ldots, k_n$ be n distinct small odd primes such that $q = 2^m k_1 k_2 k_3 \ldots k_n - 1$ is prime and E be a supersingular elliptic curve over $\mathbb{F}_q$. Suppose κ is the security parameter with $\kappa + 2 \leq m \leq \frac{1}{2}\log(q)$. The assumption states that if any PPT adversary $\mathcal{A}$ that takes E, $[\boldsymbol{a}]E$ and $([\boldsymbol{b}]E, f_{[\boldsymbol{a}][\boldsymbol{b}]E}(x(P)))$ as input where $[\boldsymbol{a}], [\boldsymbol{b}] \in cl(\mathbb{Z}[\pi_E])$, P is a point of order 2^m from $[\boldsymbol{a}][\boldsymbol{b}]E$, and outputs another pair $([\boldsymbol{b}]E, f_{[\boldsymbol{a}][\boldsymbol{b}]E}(x(P))) \neq ([\boldsymbol{c}]E, f_{[\boldsymbol{a}][\boldsymbol{b}]E}(x(Q)))$, $[\boldsymbol{c}] \in cl(\mathbb{Z}[\pi_E])$, Q is a point of order 2^m from $[\boldsymbol{a}][\boldsymbol{b}]E$, then there exists another PPT adversary $\mathcal{A}'$ that takes the same input and gets back the tuple $([\boldsymbol{c}], [\boldsymbol{c}]E, f_{[\boldsymbol{a}][\boldsymbol{b}]E}(x(Q)))$. In this assumption the randomize function f_E, $E \in cl(\mathbb{Z}(\pi_E))$ should satisfy the `Property` *3.*

2.6 SimS: An Encryption Scheme [5]

SimS = (**KeyGen**, **Encryption**, **Decryption**) is a public key encryption scheme between two parties A and B, whose message space is $\mathbb{Z}_{2^{m-2}}$, relying on the CSIDH setting. We discuss below the algorithms **KeyGen**, **Encryption** and **Decryption** of SimS in detail.

KeyGen: Let $E : y^2 = x^3 + x$ be a supersingular elliptic curve over $\mathbb{F}_q$ with Frobenius endomorphism π_E and prime $q = 2^m k_1 k_2 \dots k_n - 1$ for n distinct small odd primes $k_1, k_2, \dots, k_n$. The integer m satisfies the relation $\kappa + 2 \leq m \leq \frac{1}{2}\log(q)$. Firstly, A chooses $[\mathbf{a}] \in cl(\mathbb{Z}[\pi_E])$ and evaluates $[\mathbf{a}]E = E'$. The public key/secret key pair of A is $(PK_A, SK_A) = (E', [\mathbf{a}])$.

Encryption: To encrypt a message $Mg \in \mathbb{Z}_{2^{m-2}}$, the encryptor B first transforms Mg to an element of $\mathbb{Z}_{2^m}$ using the transformation $Mg \longrightarrow 2Mg + 1$. Next B randomly chooses $[\mathbf{b}]$ from $cl(\mathbb{Z}[\pi_E])$, and evaluates $E'' = [\mathbf{b}]E$, $E''' = [\mathbf{b}]E'$ and $P''' = [2Mg + 1]P_{E'''}$. Finally, B computes the ciphertext as $C = (E'', f_{E'''}(x(P''')))$, where $f_{E'''}$ is the randomize function induced by E'''.

Decryption: Given a ciphertext $C = (E'', f_{E'''}(x(P''')))$, the decryptor A verifies the supersingularity of E'' and determines $[\mathbf{a}]E'' = E'''$. Then A computes $x(P''')$ using the inverse mapping $f^{-1}_{E'''}$. Finally, A utilizes the Pohlig-Hellman algorithm on $x(P''')$ and $P_{E'''}$ to extract the message Mg.

2.7 IND − CPA and IND − CCA Security

An encryption scheme consists of a key generation algorithm (**KeyGen**) which generates public key/secret key pair (PK, SK), an encryption algorithm (**Encryption**) and a decryption algorithm (**Decryption**) such that **Decryption**$(SK, \textbf{Encryption}(PK, m)) = m$ for a message m. Security assumptions of an encryption scheme are depicted below.

IND − CPA (Indistinguishability under chosen plaintext attack): This is a game between a challenger ($\mathcal{C}$) and an adversary ($\mathcal{A}$).

1. ($\mathcal{C}$): Generates public key/secret key pair (PK, SK).
2. ($\mathcal{A}$): Sends encryption oracle queries to $\mathcal{C}$ a polynomial number of times.
3. ($\mathcal{A}$): Sends two plaintext messages m^0, m^1 of same length to $\mathcal{C}$. Note that m^0, m^1 should be different from the messages queried before.
4. ($\mathcal{C}$): Chooses $b \in_R \{0, 1\}$.
5. ($\mathcal{C}$): Computes **Encryption**$(PK, m^b) = C_b$ and sends it to $\mathcal{A}$.
6. ($\mathcal{A}$): Outputs a guess $b' \in \{0, 1\}$.
7. If the guess is correct then $\mathcal{A}$ wins.

$\mathsf{IND-CCA}$ (Indistinguishability under chosen ciphertext attack): This is run between a challenger ($\mathcal{C}$) and an adversary ($\mathcal{A}$).

1. ($\mathcal{C}$): Generates public key/secret key pair (PK, SK).
2. ($\mathcal{A}$): Sends decryption oracle queries to $\mathcal{C}$ a polynomially bounded number of times.
3. ($\mathcal{A}$): Sends two plaintext messages m^0, m^1 of the same length to $\mathcal{C}$. Note that m^0, m^1 should be different from the messages received during decryption oracle queries.
4. ($\mathcal{C}$): Selects $b \in_R \{0, 1\}$.
5. ($\mathcal{C}$): Determines **Encryption**$(PK, m^b) = C_b$ and sends it to $\mathcal{A}$.
6. ($\mathcal{A}$): Outputs a guess b'.
7. If the guess is correct then $\mathcal{A}$ wins.

2.8 SeaSign: A Signature Scheme [4]

$\mathsf{SeaSign}$ = (**KeyGen**, **Signature**, **Verifiaction**) is a digital signature scheme based on isogeny between elliptic curves. We describe below the algorithms of $\mathsf{SeaSign}$.

KeyGen: This algorithm is run by a signer A. Let E be a supersingular elliptic curve over $\mathbb{F}_q$ with the endomorphism π for some prime q. At first A chooses $[\mathbf{a}] \in cl(\mathbb{Z}[\pi])$, evaluates $[\mathbf{a}]E = E'$ and sets the public key/secret key pair as $(PK_A, SK_A) = (E', [\mathrm{a}])$.

Signature: To sign on a message $Mg \in \{0, 1\}^*$, A randomly selects $[\mathbf{d}_k] \in cl((Z)[\pi])$ and computes $E_k = [d_k]E$. Later on, A calculates $H(E_1, \ldots, E_t, Mg) = b_1 \parallel \ldots \parallel b_t$, where H is a cryptographic hash function and $b_i \in \{0, 1\}$. A then sets $[\mathbf{z}_k] = [\mathbf{d}_k]$ if $b_k = 0$ and $[\mathbf{z}_k] = [\mathbf{d}_k][\mathbf{a}]^{-1}$, otherwise. Finally, A outputs the signature as $\sigma = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t, b_1, b_2, \ldots, b_t)$.

Verification: On receiving the signature σ from A, the verifier determines $E'_k = [\mathbf{z}_k]E$ if $b_k = 0$ and $E'_k = [\mathbf{z}_k]E'$, otherwise. It then evaluates $H(E'_1, \ldots, E'_t, Mg) = b'_1 \parallel \ldots \parallel b'_t$ and checks the correctness of the equality $b_1 \parallel \ldots \parallel b_t = b'_1 \parallel \ldots \parallel b'_t$. If it holds then the signature is valid, otherwise it is invalid.

2.9 $\mathsf{EUF-CMA}$ Security

The security assumptions of a signature scheme consisting of the algorithms **KeyGen**, **Signature** and **Verification** is given below.

$\mathsf{EUF-CMA}$ (Existential unforgeability under chosen-message attack): A challenger ($\mathcal{C}$) and an adversary ($\mathcal{A}$) are playing the attack game as follows:

Setup: $\mathcal{C}$ runs the **KeyGen** algorithm to generate public key/signing key pair $(\mathsf{PK}, \mathsf{SK})$ and forwards PK to $\mathcal{A}$.

Sign-query: $\mathcal{A}$ makes query to $\mathcal{C}$ for signature of a message m. In the following, C runs the algorithm **Signature** and returns a valid signature σ of m to $\mathcal{A}$.

Forgery: $\mathcal{A}$ outputs a forge signature σ' for a message m' which has not been queried before in **Sign-query** phase.

If the pair (m', σ') passes the **Verification** algorithm then $\mathcal{A}$ wins the game.

2.10 Signcryption Scheme

A signcryption scheme [13, 14] consists of the following four algorithms:

pp$\longleftarrow$**Setup** (1^κ): On input of the security parameter κ, it outputs public parameter pp.

$((PK_s, SK_s), (PK_r, SK_r)) \longleftarrow$**KeyGen**$(1^\kappa, pp)$: This algorithm on input pp, outputs a public key/private key pairs (PK_s, SK_s) and (PK_r, SK_r) for sender and receiver respectively.

C$\longleftarrow$**Signcrypt**(m, SK_s, PK_r): Given a message m, sender signcrypts m with (SK_s, PK_r) and outputs a ciphertext C.

$(m \ \vee \ \perp)\longleftarrow$**Unsigncrypt**$(C, SK_r, PK_s)$: Given the ciphertext C, the receiver uses (SK_r, PK_s) to unsigncrypt C for extracting the original associated message m.

If unsigncryption is successful then it outputs $m =$ **Unsigncrypt**(C, SK_r, PK_s); otherwise, it outputs $\perp$.

2.11 Security Games for Signcryption Scheme

A signcryption scheme consisting of the algorithms (**Setup**, **KeyGen**, **Signcrypt**, **Unsigncrypt**) is a combination of signature and encryption mechanisms, and should satisfy both confidentiality and unforgeability which are discussed below.

Confidentiality $(\mathsf{IND-CCA})$: It is an attack game run between a challenger ($\mathcal{C}$) and an adversary ($\mathcal{A}$). They have the common input pp.

Initial: $\mathcal{C}$ generates public key/secret key pair $(\mathsf{PK}_r^*, \mathsf{SK}_r^*)$ and forwards (PK_r^*, pp) to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ makes a polynomially bounded unsigncryption queries for (C', PK_s) in an adaptive manner to $\mathcal{C}$ to get a valid plaintext m.

Challenge: In this phase, $\mathcal{A}$ sends $(m_0, m_1, \mathsf{PK}_s^*, \mathsf{SK}_s^*)$ to $\mathcal{C}$ for two equal length plaintext messages m_0 and m_1. $\mathcal{C}$ selects a challenge ciphertext $C_b =$ **Signcrypt**$(m_b, \mathsf{SK}_s^*, PK_r^*)$ by choosing $b \in_R \{0, 1\}$.

Guess: $\mathcal{A}$ returns a random bit b'. $\mathcal{A}$ will win if $b = b'$; otherwise, it loses.

Unforgeability ($\mathsf{EUF}-\mathsf{CMA}$): This game is played between between a challenger ($\mathcal{C}$) and an adversary ($\mathcal{A}$).

Initial: $\mathcal{C}$ runs the algorithm **KeyGen** in order to generate public key/secret key pair $(\mathsf{PK}_s^*, \mathsf{SK}_s^*)$ and forwards PK_s* to $\mathcal{A}$.

Signcryption query: In this phase, $\mathcal{A}$ makes a polynomially bounded signcryption queries with a tuple of the form (m, SK_r, PK_r) in an adaptive manner to $\mathcal{C}$ to get a valid ciphertext C=**Signcrypt**$(m, PK_r, PK_s^*, SK_s^*)$. Note that the pair ($PK_r$, SK_r) is generated by the adversary $\mathcal{A}$ for the intended receiver.

Forgery: With a fresh plaintext m^* and its ciphertext C^*, $\mathcal{A}$ outputs a tuple $(m^*, C^*, PK_r^*, SK_r^*)$ where m^* should not be used and C^* should not be received by $\mathcal{A}$ during **signcryption query** phase. If the tuple passes the verification process, i.e., **Unsigncrypt**$(C^*, SK_r^*, PK_s^*) = m^*$ then $\mathcal{A}$ wins the game, else, it loses.

The public key encryption scheme [5] possesses $\mathsf{IND}-\mathsf{CPA}$ as well as $\mathsf{IND}-\mathsf{CCA}$ security and the signature scheme [4] satisfies $\mathsf{EUF}-\mathsf{CMA}$ security. We prove that our signcryption scheme achieves both $\mathsf{IND}-\mathsf{CCA}$ and $\mathsf{EUF}-\mathsf{CMA}$ security.

2.12 Pohlig-Hellman Algorithm [9]

The discrete logarithm problem on a finite cyclic group of smooth order (that is, the order is divisible by small primes) can be efficiently solved by Pohlig-Hellman algorithm. Let $\mathbb{Z}_{2^m}$ be a finite group with x as a generator and $y = lx$ for some integer l. Then Pohlig-Hellman algorithm can efficiently compute l. In this paper, we choose the Montgomery form of an elliptic curve $E : y^2 = x^3 + ex^2 + x$ over a finite field $\mathbb{F}_q$, where $q = 2^m k_1 k_2 \dots k_n - 1$ for n distinct odd (sufficiently small) primes $k_1, k_2, \dots, k_n$. Suppose $W = \langle P \rangle$, where P is a point of order 2^m in $E(\mathbb{F}_q)$ and $Q \in \langle P \rangle$. Then the algorithm outputs l or $2^m - l$ according to whether $l < 2^{m-1}$ or $l > 2^{m-1}$, respectively.

Algorithm 1 Pohlig-Hellman algorithm on a Montgomery curve

Input: A Montgomery curve $E : y^2 = x^3 + ex^2 + x$, $e \in \mathbb{F}_q$ and x-coordinate of P of order 2^m in $E(\mathbb{F}_q)$ and x-coordinate of $Q \in \langle P \rangle$

1: $x(P_0) \longleftarrow x(P)$
2: $x(Q_0) \longleftarrow x(Q)$
3: **for each** $i \in 1, 2, \dots, m-2$ **do**
4: $\quad x(P_i) \longleftarrow x(2P_{i-1})$
5: $\quad x(Q_i) \longleftarrow x(2Q_{i-1})$
6: **end for**
7: $l \longleftarrow 1$
8: **for each** $i \in 1, 2, \dots, m-1$ **do**
9: $\quad x(T) \longleftarrow x(lQ_{m-i})$
10: $\quad$ **if** $x(P_{r-i}) \neq x(T)$ **then**
11: $\quad\quad l \longleftarrow l + 2^i$
12: $\quad$ **end if**
13: **end for**

Output: l or $2^m - l$ such that $P = lQ$

2.13 Rejection Sampling [29]

Rejection sampling is a technique, which prevents any leakage of information of the secret key. This concept was proposed by Lyumbashrvsky [29]. To understand this method let us suppose that there be two parties S and T. At first, S chooses a secret vector $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in [-k, k]^n$ where $k \in \mathbb{N}$, so that $\prod_{t=1}^{n} J_t^{a_t}$ covers almost all the ideal classes to make the output distribution uniform. After that, another random vector $\mathbf{a}' = (a'_1, a'_2, \ldots, a'_n) \in [-(nt+1)k, (nt+1)k]^n$ where $t, k \in \mathbb{N}$ is selected by S. On the other hand, T chooses a random bit $b \in \{0, 1\}$ and forwards to S. In the following, S evaluates $\chi = \mathbf{a}' - \mathbf{a}$ for $b = 0$ and $\chi = \mathbf{a}'$ for $b = 1$. To avoid any leakage of the secret key $\mathbf{a}$ we can filter χ as $\chi \in [-ntk, ntk]$. The distribution of the vector χ is uniform distribution and it is independent of the private vector $\mathbf{a}$.

3 Proposed Isogeny Based Signcryption

A high level overview: The proposed signcryption scheme is designed based on the concepts of [4, 5]. We use IBC in our construction to achieve post-quantum security. Our signature scheme involves three algorithms: KeyGen, Signcryption and Unsigncryption. During KeyGen, a sender and a receiver generate their private key/public key pair (SK_s, PK_s) and (SK_r, PK_r), respectively. Next, the sender runs Signcryption with (SK_s, PK_r) to generate a ciphertext C of a message Mg. The algorithm Unsigncryption is run by the receiver to decrypt the cipertext C using (SK_r, PK_s) and verify that the message Mg is the actual message encrypted by the sender. During Unsigncryption, the Pohlig-Hellman Algorithm 1 for the Montgomery curve has been utilized to solve a discrete logarithm problem in $\mathbb{Z}_{2^m}$. Additionally we use rejection sampling in our proposed scheme. Detail description of our scheme is provided below.

Specific point on elliptic curve:

Let us take a supersingular elliptic curve $E : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_q$, where q is prime and $q = 2^m k_1 k_2 k_3 \ldots k_n - 1$, for n distinct small odd primes $k_1, k_2, k_3, \ldots, k_n$. Indeed, $|E(\mathbb{F}_q)| = 2^m k_1 k_2 k_3 \ldots k_n$ and there exits points of orders divided by 2^m. Among them, we consider a point P'_E from $E(\mathbb{F}_q)$ of order divisible by 2^m with largest x-coordinate in $\{-2, -3, -4, \ldots, -q+1\}$ and we build a point P_E as follows.

$$P_E = k_1 k_2 \ldots k_n P'_E. \tag{3.1}$$

Protocol 1. *Proposed signcryption scheme*

$pp \longleftarrow$ Setup (1^κ): It outputs the public parameter $pp=(\mathcal{E}, q)$, on input of a security parameter κ where $\mathcal{E} : y^2 = x^3 + x$ is a supersingular elliptic curve over $\mathbb{F}_q$ with prime $q = 2^m k_1 k_2 \ldots k_n - 1$ for n distinct small odd primes $k_1, k_2, \ldots, k_n$ and $\kappa + 2 \leq m \leq \frac{1}{2}\log(q)$. Let k be a positive integer.

$((SK_s, PK_s), (SK_r, PK_r)) \longleftarrow$ KeyGen (pp): On input pp, each of the sender and receiver generates public key/secret key pair as follows.

1. *Sender randomly chooses $\boldsymbol{a} = (a_1, a_2, \ldots, a_n) \in [-k, k]^n$, computes $\mathcal{E}_1 = [\boldsymbol{a}]\mathcal{E}$, keeps $SK_s = \boldsymbol{a}$ as secret and outputs $PK_s = \mathcal{E}_1$ as public key.*
2. *Receiver selects $\boldsymbol{b} = (b_1, b_2, \ldots, b_n) \in [-k, k]^n$ in a random manner, computes $\mathcal{E}_2 = [\boldsymbol{b}]\mathcal{E}$, retains $SK_r = \boldsymbol{b}$ as secret and publishes $PK_r = \mathcal{E}_2$ as public key.*

$C \longleftarrow$ Signcryption(PK_r, SK_s, Mg): Sender does the following operations to get a valid signcryption on a plaintext message $Mg \in \{0, 1\}^{m-2}$:

1. *For* $k = 1, 2, \ldots, t$, *chooses* $\boldsymbol{d}_k = (d_1^k, d_2^k, \ldots, d_n^k) \in [-(nt+1)k, (nt+1)k]^n$ *and evaluates* $E_k = [\boldsymbol{d}_k] * \mathcal{E}$.
2. *Calculates* $H(E_1, \ldots, E_t, Mg) = b_1 \parallel \ldots \parallel b_t$, *where* $H : \{0,1\}^{t\log(p)+(m-2)} \longleftarrow \{0,1\}^t$ *is a cryptographically secure collision resistant hash function.*
3. *For* $k = 1, 2, \ldots, t$, *sets* $\boldsymbol{z}_k = \boldsymbol{d}_k$ *if* $b_k = 0$ *and otherwise, sets* $\boldsymbol{z}_k = \boldsymbol{d}_k - \boldsymbol{a}$.
4. *If* $\boldsymbol{z}_k \in [-ntk, ntk]^n$ *then performs the next step; otherwise, starts from step 1.*
5. *Evaluates* $[\boldsymbol{a}]\mathcal{E}_2 = \mathcal{E}_3$.
6. *Maps* $Mg \in \{0,1\}^{m-2}$ *to an element* $M' \in \mathbb{Z}_{2^m}^*$ *by taking* $M' = 2Mg + 1$.
7. *Computes the point* $P_{\mathcal{E}_3}$ *as mentioned in Equation 3.1 and the function* $f_{\mathcal{E}_3}$ *as discussed in Section 2.4.*
8. *Calculates* $P_3 = [2Mg+1]P_{\mathcal{E}_3} = (x(P_3), y(P_3))$ *and* $f_{\mathcal{E}_3}(x(P_3)) = \eta$.

Sender outputs the ciphertext as $C = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t, b_1, \ldots, b_n, \eta)$. *Refer to Figure 1 for the flowchart of the* Signcryption *Algorithm.*

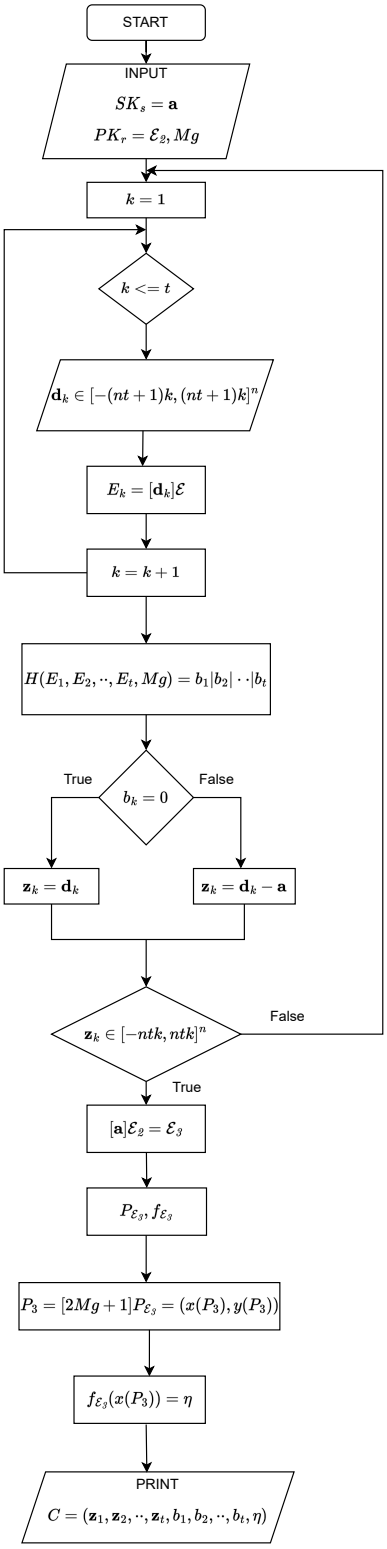
$Mg \longleftarrow$ Unsigncryption(PK_s, SK_r, C): *On receiving a ciphertext* C, *the receiver executes the following steps to obtain a verified plaintext message*:

1. *Computes* $[\boldsymbol{b}]\mathcal{E}_1 = \mathcal{E}_3$.
1. *Evaluates the function* $f_{\mathcal{E}_3}$, *its inverse* $f_{\mathcal{E}_3}^{-1}$ *and the point* $P_{\mathcal{E}_3}$ *by a similar method as mentioned in step* 6 *of* Signcryption.
2. *Calculates* $f_{\mathcal{E}_3}^{-1}(\eta) = x(P_3)$, *where* $x(P_3)$ *is the* x*-coordinate of the point* P_3 *on* $\mathcal{E}_3$.
3. *Using the Pohlig-Hellman Algorithm 1, the receiver computes* M' *from* $x(P_3)$ *and* $P_{\mathcal{E}_3}$. *If* $2^{m-1} < M$, *it computes* $2^m - M'$.
4. *Calculates* $\frac{M'-1}{2} = Mg$.
5. *For* $k = 1, 2, \ldots, t$, *computes* $E_k = [\boldsymbol{z}_k]\mathcal{E}$ *if* $b_k = 0$ *and* $E_k = [\boldsymbol{z}_k]\mathcal{E}_1$ *if* $b_k = 1$.
6. *Determines* $H(E_1, \ldots, E_t, Mg) = b_1' \parallel \ldots \parallel b_t'$.
7. *If* $(b_1', \ldots, b_t') = (b_1, \ldots, b_t)$ *then the signcryption is valid and otherwise, it returns invalid.*

Refer to Figure 1 for the flowchart of the Unsigncryption *Algorithm.*

Correctness: In the proposed scheme, $\mathcal{E}_1 = [\mathbf{a}]\mathcal{E}$ and $\mathcal{E}_2 = [\mathbf{b}]\mathcal{E}$. Indeed, $[\mathbf{a}]\mathcal{E}_2 = [\mathbf{a}][\mathbf{b}]\mathcal{E} = [\mathbf{b}][\mathbf{a}]\mathcal{E} = [\mathbf{b}]\mathcal{E}_1 = \mathcal{E}_3$. As a consequence, both sender and receiver get the curve $\mathcal{E}_3$ which they can use to implement function $f_{\mathcal{E}_3}^{-1}$ on some element of $\mathbb{F}_q$. On receiving the ciphertext C from the sender, the receiver will evaluates $f_{\mathcal{E}_3}^{-1}(\eta)$ and finds the x-coordinate of P_3. Also, the receiver is able to compute the specific point $P_{\mathcal{E}_3}$. Using the Pohlig-Hellman Algorithm 1 on the x-coordinates of $P_{\mathcal{E}_3}$ and P_3, the receiver can efficiently compute the message Mg.
In step 6 of Unsigncryption, the receiver does some class group operations for verifying the validity of the encrypted message. The correctness of those operations are discussed below:

$$E_k = \begin{cases} [\mathbf{z}_k]\mathcal{E} = [\mathbf{d}_k]\mathcal{E}, & \text{if } b_k = 0 \\ [\mathbf{z}_k]\mathcal{E}_1 = [\mathbf{d}_k - \mathbf{a}]\mathcal{E}_1 = [\mathbf{d}_k]\mathcal{E}, & \text{if } b_k = 1. \end{cases}$$



(a) Flowchart of Signcryption algorithm **(b)** Flowchart of Unsigncryption algorithm

Figure 1 Flowchart of Signcryption and Unsigncryption Algorithm

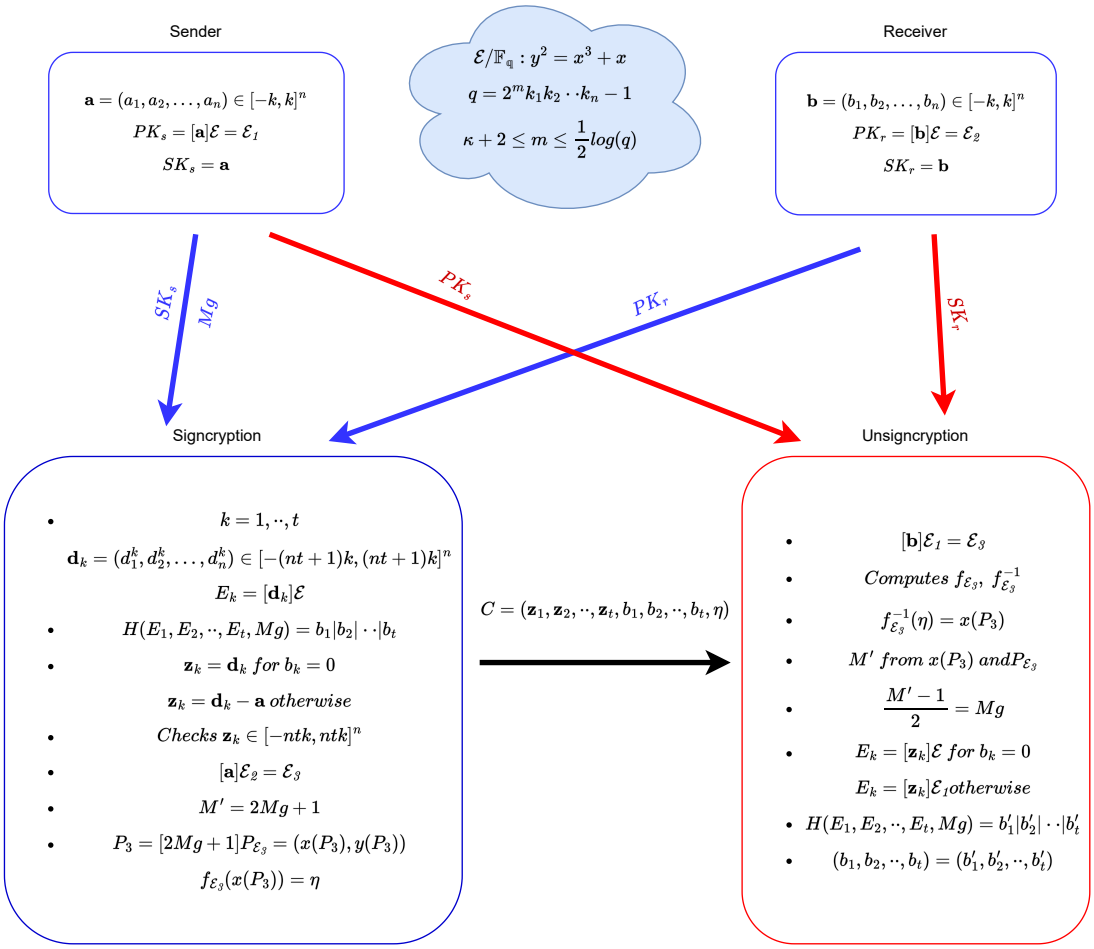


Figure 2 Proposed signcryption scheme

4 Security

In this section we show that the signcryption scheme achieves both $\mathsf{IND} - \mathsf{CCA}$ and $\mathsf{EUF} - \mathsf{CMA}$ security. Security properties of the underlying encryption [5] and signature scheme [4] are given in the following three theorems:

Theorem 4.1. *[5] If* CSSIDDH *assumption holds and* f_E *satisfies the property (P2), then the encryption scheme* Sims *is* $\mathsf{IND} - \mathsf{CPA}$ *secure.*

Theorem 4.2. *[5] If* CSSIDDH, CSSIKOE *assumptions hold and* f_E *satisfies the property (P2), then* Sims *is* $\mathsf{IND} - \mathsf{CCA}$ *secure.*

Theorem 4.3. *[4] Under the hardness assumption of* GAIP, *the signature scheme* SeaSign *is* $\mathsf{EUF} - \mathsf{CMA}$ *secure.*

Theorem 4.4. *Taking into account the* $\mathsf{IND} - \mathsf{CCA}$ *security of* Sims, *our signcryption scheme is* $\mathsf{IND} - \mathsf{CCA}$ *secure.*

Proof. We prove that our proposed signcryption is $\mathsf{IND} - \mathsf{CCA}$ secure by contradiction. By absurd, we assume that the scheme is not $\mathsf{IND} - \mathsf{CCA}$ secure. As a result there should be

a PPT adversary $\mathcal{A}_1$ against the $\mathsf{IND}-\mathsf{CCA}$ game of signcryption scheme who has access to the unsigncryption oracle and is able to decide whether a given ciphertext is the signcryption of m_0 or m_1 with non negligible advantage. Initially, the challenger ($\mathcal{C}$) runs **KeyGen** to compute its own private key/secret key pair (SK_r^*, PK_r^*), where $SK_r^* = \mathbf{b} = (b_1, b_2, \ldots, b_n)$ and $PK_r^* = [\mathbf{b}]\mathcal{E} = E_2$ and forwards E_2 to $\mathcal{A}_1$. Suppose $\mathcal{A}_1$ makes some unsigncryption queries with (C_1, PK_s^1), (C_2, PK_s^2), (C_3, PK_s^3),..., (C_n, PK_s^n), where $C_i = (\mathbf{z}_1^i, \mathbf{z}_2^i, \ldots, \mathbf{z}_t^i, b_1^i, b_2^i, \ldots, b_t^i, \eta_i)$, $i = 1, 2, \ldots, n$.
Let $\mathcal{A}_2$ be an adversary against the $\mathsf{IND}-\mathsf{CCA}$ game of the underlying encryption scheme. Initially, $\mathcal{A}_2$ takes the pairs (η_i, PK_s^i), $i = 1, 2, \ldots, n$, from $\mathcal{A}_1$. As $\mathsf{CSSIKOE}$ assumption holds, there exists another PPT adversary $\mathcal{A}_3$ which outputs the tuple $(\eta_i, PK_s^i, [\mathbf{a}]^i])$ for $i = 1, 2, \ldots, n$ in parallel such that $PK_s^i = [\mathbf{a}]^i]\mathcal{E}$. After getting $[\mathbf{a}]^i]$, $\mathcal{A}_3$ can decrypt any ciphertext of the underlying encryption scheme to extract the associated message, which is also message of the signcryption scheme. Let $\mathcal{A} = (\mathcal{A}_2, \mathcal{A}_3)$. Then without using any decryption oracle, $\mathcal{A}$ can decide whether the target ciphertext is the encryption of m_0 or m_1. As a result, the underlying encryption scheme does not remain $\mathsf{IND}-\mathsf{CPA}$ secure. However, from the theorem 4.1 we can say that the encryption scheme Sims is $\mathsf{IND}-\mathsf{CPA}$ secure. This is a contradiction. So we can conclude that our signcryption scheme is $\mathsf{IND}-\mathsf{CCA}$ secure. □

Theorem 4.5. *Under the hypothesis that the* GAIP *assumption holds, our signcryption scheme is* $\mathsf{EUF}-\mathsf{CMA}$ *secure.*

Proof. We prove it by contradiction. If possible let there be a PPT adversary $\mathcal{A}$ who can break the $\mathsf{EUF}-\mathsf{CMA}$ security of our signcryption scheme by generating a forge signcrypted ciphertext against our proposed signcryption scheme. Then we show that it is possible to design a simulator ($\mathcal{S}^{\mathcal{A}}$) with the help of $\mathcal{A}$ to break the security of the underlying signature scheme. As the proposed signcryption scheme is assumed not to be $\mathsf{EUF}-\mathsf{CMA}$ secure for a PPT adversary $\mathcal{A}$, the following steps will be performed to forge a valid ciphertext by $\mathcal{A}$ with $\mathcal{C}$:

- **Initial**: Challenger $\mathcal{C}$ runs **KeyGen** to generate its own private key/secret key pair (PK_s^*, SK_s^*), where $SK_s^* = \mathbf{a}$, $PK_s^* = [\mathbf{a}]E$ and sends PK_s^* to $\mathcal{A}$.
- **Signcrypt query**: In this phase, $\mathcal{A}$ can perform a polynomially bounded signcryption query as follows: $\mathcal{A}$ submits (Mg, SK_r, PK_r) to $\mathcal{C}$, where PK_r = intended receiver's public key and SK_r = intended receiver's secret key. Here $SK_r = \mathbf{b}$ and $PK_r = [\mathbf{b}]E$. On receiving (Mg, SK_r, PK_r), $\mathcal{C}$ runs the $\mathsf{Signcryption}$ algorithm with its own secret key SK_s^* and outputs a valid ciphertext C'.
- **Forgery**: $\mathcal{A}$ outputs a new public key/private key pair (PK_r^*, SK_r^*) and a valid ciphertext $C^* = \mathsf{Signcryption}(PK_r^*, SK_s^*, Mg^*) = (\mathbf{z}_1', \mathbf{z}_2', \ldots, \mathbf{z}_t', b_1', b_2', \ldots, b_t', x_1')$ on a message Mg^* which has not been queried before, where $SK_r^* = \mathbf{d}$ and $PK_r^* = [\mathbf{d}]E$ (say).

Now since C^* is a valid ciphertext of the signcryption scheme, then $(\mathbf{z}_1', \mathbf{z}_2', \ldots, \mathbf{z}_t', b_1', b_2', \ldots, b_t')$ is a valid signature on Mg^* produced by $\mathcal{A}$. As a result we can design a simulator $\mathcal{S}^{\mathcal{A}}$ with the help of $\mathcal{A}$ to generate a valid signature on Mg^* for the underlying signature scheme [4] is broken. However the signature scheme of [4] attains $\mathsf{EUF}-\mathsf{CMA}$ security under the GAIP assumption by Theorem 4.3. This leads to a contradiction. Therefore, our signcryption scheme is $\mathsf{EUF}-\mathsf{CMA}$ secure. □

5 Efficiency

In this section, we discuss the communication and computation costs of our proposed scheme. As $|cl(Z[\pi])| = \#\{[J_1]^{a_1}, [J_2]^{a_2}, \ldots, [J_n]^{a_n} : a_1, a_2, \ldots, a_n \in \{-k, \ldots, k\}\}$, we can infer that a random element in $cl(Z[\pi])$ is of size $n\log(2k+1)$ bit. In order to compute the ciphertext in [5], the authors used two class group actions, whereas we use only one class group action in the encryption portion. Here we require the following SiGamal primes [9]:

p_{128}: The 522 bit prime p_{128} is denoted by $2^{130}k_1k_2\ldots k_{60}-1$. Here, $k_1 < k_2 < \ldots < k_{60} = 569$ are therefore small primes. The key bound of p_{128} is taken as 10. Set the point $P_{128} = k_1k_2\ldots k_{60}\bar{P}_{128} \in \mathcal{E}(\mathbb{F}_{p_{128}})$ of order 2^{130}, where the x-component of the point $\bar{P}_{128}$ is 331.

p_{256}: $p_{256} = 2^{258}k_1k_2\ldots k_{43}-1$ is a 515 bit prime with small odd primes $k_1 < k_2 < \ldots < k_{43} = 307$. The key bound of p_{256} is taken as 32. The point $P_{256} = k_1k_2\ldots k_{43}\bar{P}_{256} \in \mathcal{E}(\mathbb{F}_{p_{256}})$ of order 2^{258}, where the x-component of the point $\bar{P}_{256}$ is 199.

To design the signcryption scheme, we use the encryption scheme SimS whose computational costs is given in Table 1 for p_{128} and p_{256} in terms of field multiplication (M).

Table 1 Computational costs of calculating a class group action in terms of field multiplication (M) for SimS

SiGamal primes	p_{128}	p_{256}
One class group action	576124 M	1023400 M
KeyGen	576124 M	1023827 M
Encryption	1159533 M	2057297 M
Decryption	679733 M	1417401 M

According to SimS, to encrypt a message, the encryptor B has used two class group operations while to decrypt the message A has employed only one class group operation. We have used $t+1$ class group operations in Signcryption as well as in Unsigncryption to create our proposed signcryption scheme. The computational cost of our signcryption scheme is less than the sum of the computational costs of the basic encryption scheme Sims [5] and signature scheme SeaSign [4], which we have used to construct our signcryption scheme. Cost of calculating one class group action is given in Table 1. Rest of the computational costs are same as given in [5]. To calculate the actual running time complexity of our proposed design, we also implemented our scheme using the free and open source mathematics software SageMath[1]. The specific experimental scenarios (hardware and software specifications) for performing experiments is given in Table 2. Results of our experiments are documented in Table 3.

We compute the sizes of keys, ciphertext (encryption), signature , and ciphertext (signcryption) which are given in Table 5. To achieve NIST level 1 security we compute the computational costs communication costs of our scheme in Table 4 and Table 6 respectively for p_{128} and p_{256}. We have plotted computational costs of class group operation and Signcryption algorithm in Figure 3.

[1]https://www.sagemath.org.

Table 2 Specific experimental scenarios for the implementation

Software	SageMath (v9.2.1)
Language	Python
Processor	Dell Intel Core i9
RAM	32 GB
Operating System	Linux Lite (v5.2)

Table 3 Actual running time complexity our scheme (in seconds)

	p_{128}	p_{256}
Signcryption cost	2580	10280
Unsigncryption cost	2580	10280

Table 6 Communication cost

SiGamal primes	p_{128}	p_{256}
Ciphertext (encryption)	130.5 bytes	128.5 bytes
Signature	16556.8 bytes	12693.08 bytes
Ciphertext (encryption)+Signature	16687.3 bytes	12821.83 bytes
Ciphertext (signcryption)	16622.05 bytes	12757.45 bytes

The authors [1] of CSIDH suggested three parameter setting for NIST security classification [21]. To achieve NIST level 1, NIST level 3 and NIST level 5 security, they proposed CSIDH-512, CSIDH-1024, CSIDH-1792, respectively [4]. Some authors [23, 24] has suggested that the CSIDH is failed to achieve NIST level 1 security and turn down the quantum security properties.
In [22], Chávez-Saab et al. proposed some large CSIDH parameters for NIST level 1, NIST level 2 and NIST level 3. From those we discuss the computation costs for CSIDH-3072, CSIDH-5120, CSIDH $-$ 8192 in terms of field multiplication(M). We take $t = 128$ for CSIDH $-$ 3072, $t = 192$ for CSIDH $-$ 5120 and $t = 256$ for CSIDH $-$ 8192. $t + 1$ class group operations has been used in our Signcryption algorithm as well as in Unsigncryption. Therefore, total class group operation cost for Signcryption algorithm as well as Unsigncryption algorithm will be of 114,532,650 M for CSIDH $-$ 3072, 198,741,750 M for CSIDH $-$ 5120, and 506,058,700 M for CSIDH $-$ 8192. Here we use the MCR-style [20] class group computation and the secret key bound m for each class group computation is taken as 1. We refer to Table 7 for the sizes of ciphertext (encryption), signature and ciphertext (signcryption), corresponding to the different CSIDH variants.

6 Conclusion

In this work, we proposed an isogeny based signcryption scheme. We used the SimS [5] and SeaSign [4] as the underlying building blocks of our design. The proposed scheme attains

Table 4 Computational costs of our scheme in terms of field multiplication (M)

Cost	p_{128}	p_{256}
Signcryption cost	74327281 M	132029097 M
Encryption + Signature cost	74903405 M	133052417 M
Unsigncryption cost	74423605 M	132412601 M
Decryption + Verification cost	74423605 M	132412601 M

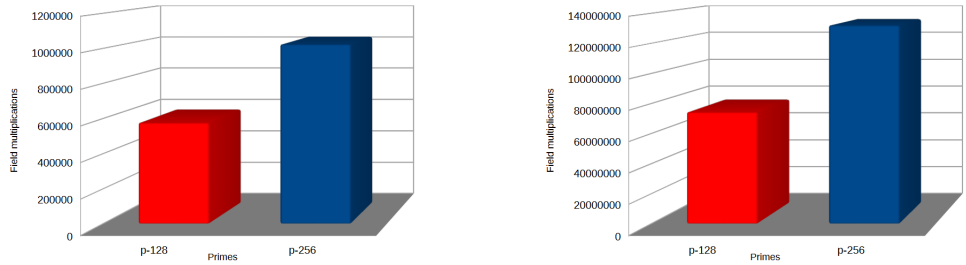


Figure 3 Computational efficiency with respect to class group operations and Signcryption for p_{128} and p_{256}

$\mathsf{IND}-\mathsf{CCA}$ and $\mathsf{EUF}-\mathsf{CMA}$ security under the hardness assumptions: CSSIDDH, GAIP and CSSIKOE. The ciphertext size of the proposed signcryption scheme is less than the sum of the individual signature size and ciphertext (encryption) size of the underlying schemes. A similar conclusion holds for the computation cost. The ciphertext size and each public key size of our signcryption turns out to be $nt\log(2k+1)+t+\log(q)$ and $\log(p)$ to achieve κ bits of security level. From Kuperberg's algorithm, we can state that to make our scheme quantum computer secure we have to assume $q < 2^{2\kappa^2}$ on the input of security parameter κ. In particular, our signcryption scheme is the first of its kind in the context of isogeny based cryptography.

References

[1] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes. 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.

[2] J. M. Couveignes. Hard homogeneous spaces. *IACR Cryptol. ePrint Arch.*, 2006:291, 2006.

[3] L. De Feo. Mathematics of isogeny based cryptography. *arXiv preprint arXiv:1711.04062*, 12, 2017.

[4] L. De Feo and S. D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 759–789. Springer, 2019.

[5] T. B. Fouotsa and C. Petit. SimS: A simplification of SiGamal. In *International Conference on Post-Quantum Cryptography*, pp. 277–295. Springer, 2021.

[6] F. Li, F. T. B. Muhaya, M. K. Khan, and T. Takagi. Lattice-based signcryption. *Concurrency and Computation: Practice and Experience*, 25(14):2112–2122, 2013.

Table 5 Sizes of keys, ciphertext (encryption), Signature and ciphertext (signcryption)

Sender's public key (signcryption)	$\log(q)$ bit
Sender's Secret key (signcryption)	$n\log(2k+1)$ bit
Receivers's public key (signcryption)	$\log(q)$ bit
Receivers's secret key (signcryption)	$n\log(2k+1)$ bit
Ciphertext (signcryption)	$nt\log(2ntk+1)+t+\log(q)$ bit
Ciphertext (encryption) [5]+Signature [4]	$nt\log(2ntk+1)+t+2\log(q)$ bit

Table 7 Sizes of ciphertext (encryption), signature and ciphertext (signcryption)

CSIDH variants	Ciphertext (encryption)	Signature	Ciphertext (encryption)+Signature	Ciphertext (signcryption)
CSIDH − 512	128 bytes	4127.5 bytes	4255.5 bytes	4191.25 bytes
CSIDH − 1024	254.7 bytes	17112.9 bytes	17367.6 bytes	17249.4 bytes
CSIDH − 1792	446.5 bytes	51421.7 bytes	51868.2 bytes	51644.9 bytes
CSIDH − 3072	768 bytes	85297.6 bytes	86065.6 bytes	85681.6 bytes
CSIDH − 5120	1280 bytes	212429.76 bytes	213709.76 bytes	213069.76 bytes
CSIDH − 8192	2048 bytes	449629.44 bytes	451677.44 bytes	450653.44 bytes

[7] X. Lu, Q. Wen, Z. Jin, L. Wang, and C. Yang. A lattice-based signcryption scheme without random oracles. *Frontiers of Computer Science* 8(4):667–675, 2014.

[8] J. Malone-Lee and W. Mao. Two birds one stone: signcryption using RSA. In *Cryptographers' Track at the RSA Conference*, pp. 211–226. Springer, 2003.

[9] T. Moriya, H. Onuki, and T. Takagi. SiGamal: A supersingular isogeny-based PKE and its application to a PRF. In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 551–580. Springer, 2020.

[10] P. W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review* 41(2):303–332, 1999.

[11] R. Steinfeld and Y. Zheng. A signcryption scheme based on integer factorization. In *International Workshop on Information Security*, pp. 308–322. Springer, 2000.

[12] Z. Wang, Y.-L. Han, W.-C. Liu, and L. Chen. Anti-quantum generalized signcryption scheme based on multivariate and coding. In *2019 Chinese Control And Decision Conference (CCDC)*, pp. 3587–3594. IEEE, 2019.

[13] J. Yan, L. Wang, L. Wang, Y. Yang, and W. Yao. Efficient lattice-based signcryption in standard model. *Mathematical Problems in Engineering* 2013:702539, 2013.

[14] X. Yang, H. Cao, W. Li, and H. Xuan. Improved lattice-based signcryption in the standard model. *IEEE Access* 7:155552–155562, 2019.

[15] D. H. Yum and P. J. Lee. New signcryption schemes based on KCDSA. In *International Conference on Information Security and Cryptology*, pp. 305–317, Springer, 2001.

[16] Y. Zheng. Digital signcryption or how to achieve cost (signature & encryption)« cost (signature)+ cost (encryption). In *Annual International Cryptology Conference*, pp. 165–179. Springer, 1997.

[17] Abouelseoud, Yasmine. A Tripartite Signcryption Scheme with Applications to E-Commerce. In *International Journal of Computer Applications*, pp. 8887. Citeseer, 2013.

[18] Elkamchouchi, Hassan M and Abou Elkheir, Eman F and Abouelseoud, Yasmine. An Efficient Off-Line E-Cash System based on Signcryption without Bilinear Pairings. In *International Journal of Computer Applications*, 91(15). Citeseer, 2014.

[19] Meshram, Chandrashekhar and Imoize, Agbotiname Lucky and Aljaedi, Amer and Alharbi, Adel R and Jamal, Sajjad Shaukat and Barve, Sharad Kumar. An Efficient Electronic Cash System Based on Certificateless Group Signcryption Scheme Using Conformable Chaotic Maps. In *Sensors*, 21(21):7039. Multidisciplinary Digital Publishing Institute, 2021.

[20] Meyer, Michael, and Steffen Reith. A faster way to the CSIDH. In *International Conference on Cryptology in India*, pp. 137–152. Springer, 2018.

[21] National Institute of Standards and Technology: Announcing request for nominations for public-key post-quantum cryptographic algorithms (2016), https://www.federalregister.gov/d/2016-30615.

[22] Chávez-Saab, Jorge and Chi-Domínguez, Jesús-Javier and Jaques, Samuel and Rodríguez-Henríquez, Francisco. The SQALE of CSIDH: sublinear Vélu quantum-resistant isogeny action with low exponents. In *Journal of Cryptographic Engineering*, pp. 1–20. Springer, 2021.

[23] Bonnetain X, Schrottenloher A. Quantum security analysis of CSIDH. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 493–522. Springer, 2020.

[24] Peikert C. He gives C-sieves on the CSIDH. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 463–492. Springer, 2020.

[25] De Feo L, Kohel D, Leroux A, Petit C, Wesolowski B. SQISign: compact post-quantum signatures from quaternions and isogenies. In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 64–93. Springer, 2020.

[26] Beullens W, Kleinjung T, Vercauteren F. CSI-FiSh: efficient isogeny based signatures through class group computations. In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 227–247. Springer, 2019.

[27] Galbraith SD, Petit C, Silva J. Identification protocols and signature schemes based on supersingular isogeny problems. In *International conference on the theory and application of cryptology and information security*, pp. 3–33. Springer, 2017.

[28] Rostovtsev A, Stolbunov A. Public-key cryptosystem based on isogenies. In *Cryptology ePrint Archive*, 2006.

[29] Lyubashevsky V. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 598–616. Springer, 2009.