Post-Quantum Variants of ISO/IEC Standards: Compact Chosen Ciphertext Secure Key Encapsulation Mechanism from Isogenies

Kazuki Yoneyama

Ibaraki University
kazuki.yoneyama.sec@vc.ibaraki.ac.jp

Abstract. ISO/IEC standardizes several chosen ciphertext-secure key encapsulation mechanism (KEM) schemes in ISO/IEC 18033-2. However, all ISO/IEC KEM schemes are not quantum resilient. In this paper, we introduce new isogeny-based KEM schemes (i.e., CSIDH-ECIES-KEM and CSIDH-PSEC-KEM) by modifying Diffie-Hellman-based KEM schemes in ISO/IEC standards. The main advantage of our schemes are compactness. The key size and the ciphertext overhead of our schemes are about five times smaller than these of SIKE-KEM which is submitted to NIST's post-quantum cryptosystems standardization.

Keywords: Post-quantum cryptosystems, Isogeny-based cryptosystems, CSIDH

1 Introduction

1.1 Isogeny-based Cryptosystems

Post-quantum cryptosystems (PQC) are one of hottest research topics in cryptography due to the emerging of quantum computers. Though the most studied PQC is lattice-based, other alternatives are also required for risk diversification as NIST's PQC standardization [NIST]. Isogeny-based cryptosystems are one of candidates of PQC. Given two elliptic curves $E, E'/\mathbb{F}_p$, non-zero homomorphism $\psi : E \to E'$ is called an isogeny. By Vélu's formula [Vél71], given an elliptic curve E and a point R on E, we can efficiently compute an isogeny $\psi : E \to E/\langle R \rangle$ with kernel $\langle R \rangle$ by choosing an appropriate parameter. On the other hand, given two isogenous elliptic curves E and E', to find (a compact representation of) isogeny $\psi : E \to E'$ (the isogeny computation problem) is believed to be hard even for quantum computers. Isogeny-based cryptosystems rely on the isogeny computation problem and its derivations. The advantage of isogeny-based cryptosystems against other PQC candidates is compactness of the key size and the ciphertext size.

Couveignes [Cou06] initiated the research of isogeny-based cryptography by formulating the basic notion of *hard homogeneous spaces (HHSs)* which is an abstract form of isogeny graphs and class groups of endomorphism rings of (ordinary) elliptic curves. Rostovtsev and Stolbunov [RS06] proposed a DH type

key exchange scheme from ordinary elliptic curve isogenies. On the other hand, Childs et al. [CJS14] showed that the isogeny computation problem on ordinary elliptic curve isogenies can be analysed in quantum subexponential time. Then, Jao et al. [JF11] proposed supersingular isogeny-based DH type key exchange (SIDH) scheme because no quantum subexponential time analysis is known for the isogeny computation problem on supersingular elliptic curve isogenies. It is known that j-invariants $j(E) = j(E')$ (where $j(E)$ is deterministically derived from E) iff elliptic curves E and E' are isomorphic. SIDH uses this property to share j-invariants as the common session key between parties. Also, Castryck et al.[CLM$^+$18] proposed a new HHS-based key exchange scheme called *CSIDH (commutative SIDH)*, which is constructed from a group action on the set of supersingular elliptic curves defined over a prime field. Since the group action is commutative in CSIDH, we can deal with it as a similar manner to classical DH key exchange. In CSIDH, a common secret curve is obtained between parties resulting from the group action, and the Montgomery coefficient of the curve is shared as the common session key. Moreover, validity of public keys can be efficiently verified while SIDH has no efficient method yet. Hence, CSIDH is very compatible to classical DH.

There is a trade-off between the SIDH system and the CSIDH system. The advantage of SIDH is that computational time is relatively faster than the CSIDH while it is slower than other PQC candidates. For the security level corresponding to 64 bit quantum security and 128 bit classical security (i.e., NIST category 1 [NIST]), computational time for the SIDH key exchange is about 10 times faster than the CSIDH key exchange. On the other hand, the advantage of CSIDH is that the key size is more compact than SIDH while the key size of SIDH is also more compact than other PQC candidates. For the parameter of NIST category 1, the key size is about one fifth of these of SIDH. Also, another major advantage of CSIDH is efficient public key validation and applicability to non-interactive key exchange.

For NIST's PQC competition, a key encapsulation mechanism (KEM) scheme based on SIDH, called SIKE-KEM [SIKE17], was submitted as the only isogeny-based submission, and it now survives at the second round. SIKE-KEM satisfies chosen ciphertext (CCA) security in the classical random oracle (RO) model. It is obtained by applying a generic construction [HHK17] to a chosen plaintext (CPA) secure public key encryption (PKE) scheme, called SIKE-PKE, which is an extension of the hashed ElGamal PKE to SIDH-based. SIKE-KEM achieves the most compact key size and ciphertext length among NIST PQC submissions. However, the key size and the ciphertext size are still relatively large compared to classical DH-based KEM schemes. For example, SIKE-KEM needs the public key of 2640 bit and the ciphertext of 3152 bit for the parameter of NIST category 1, but ECIES-KEM [ISO] only needs the public key of 256 bit for the same level of classical security. Hence, it is an interesting question how compact we can achieve CCA-secure post-quantum KEM.

1.2 ISO/IEC Standards KEM

ISO/IEC standardizes several CCA-secure KEM schemes in ISO/IEC 18033-2 [ISO]. Such standards are important in the real world because it is helpful to implement a KEM scheme in an IT system by non-specialist engineers. Since ISO/IEC standards have been implemented in various systems and many theoretical and implementation attacks would be examined, the structures and security of these schemes are reliably sound by its maturity.

In ISO/IEC 18033-2, four DH-based schemes, two RSA-based schemes and a factoring-based scheme are standardized. In this paper, we focus on DH-based schemes (ECIES-KEM, PSEC-KEM, ACE-KEM and FACE-KEM). All these schemes satisfy CCA-security. ECIES-KEM and PSEC-KEM are proved in the RO model, and ACE-KEM and FACE-KEM are proved in the standard model (i.e., without ROs). The merit of ECIES-KEM and PSEC-KEM is compactness. For example, for 128 bit security, the public key size is 256 bit, and the ciphertext size (overhead) is 256 bit for ECIES-KEM and 384 bit for PSEC-KEM.

However, all KEM schemes in ISO/IEC 18033-2 are not quantum-resilient. If quantum computers become practical, it is known that underlying DH, RSA and factoring problems are solved in a quantum polynomial-time by Shor's algorithm [Sho94] and its variant. It is desirable to construct post-quantum KEM schemes without changing structures of standards.

1.3 Our Contribution

We give a post-quantum variants of CCA-secure KEM schemes standardized in ISO/IEC. Our scheme inherits the same structures as original KEM schemes; and thus, it is structurally sound by maturity of standards. Specifically, we extend PSEC-KEM and ECIES-KEM to CSIDH-based, called CSIDH-PSEC-KEM and CSIDH-ECIES-KEM. We use group action operations of the CSIDH system instead of elliptic curve scalar multiplications of original schemes. Thanks to compatibility of CSIDH to DH, we can retain original structures.

Also, since we can consider DH-like hard problems in the CSIDH system, CCA-security can be proved by a similar manner as original proofs in the RO model. On the other hand, in the quantum setting, an adversary poses quantum queries to ROs. It means that security proofs in the quantum RO (QRO) model are desirable. Hence, we show the security proof of CSIDH-PSEC-KEM in the QRO model. Proofs in the QRO model have several difficulties which do not happen in the RO model. Specifically, a simulator cannot extract a quantum hash query (i.e., a quantum superposition) from the hash list due to the no-cloning theorem, and must generate random values for exponentially many positions in order to simulate outputs of the hash function. We solve these hurdles by applying Zhandry's simulation technique [Zha12] to our security proof. Due to the former difficulty of the QRO model, we prove security of CSIDH-PSEC-KEM under the decisional DH-like (CSI-DDH) assumption while PSEC-KEM can be proved under the computational DH assumption. For CSIDH-ECIES-KEM, it is not easy to prove security in the QRO model because the proof needs the

gap DH-like (CSI-GDH) assumption which is not comparable to the decisional DH-like assumption.

The main advantage of our schemes are compactness. CSIDH-ECIES-KEM has the public key of 512 bit and the ciphertext of 512 bit for the parameter of NIST category 1.[1] It is just twice of original ECIES-KEM. Also, CSIDH-PSEC-KEM has the public key of 512 bit and the ciphertext of 640 bit for the parameter of NIST category 1. It is also comparable to classical DH-based KEM schemes while security can be proved in the QRO model. A detailed efficiency estimation is given in Section 5.

To summarize, our contribution is as follows:

- We introduce CSIDH-PSEC-KEM by extending PSEC-KEM to CSIDH-based. CCA-security is proved in the QRO model under the CSI-DDH assumption. It has the public key of 512 bit and the ciphertext of 640 bit for the parameter of NIST category 1. CSIDH-PSEC-KEM is the most compact post-quantum CCA-secure KEM in the QRO model as far as we know.
- We introduce CSIDH-ECIES-KEM by extending ECIES-KEM to CSIDH-based. CCA-security is proved in the RO model under the CSI-GDH assumption. It has the public key of 512 bit and the ciphertext of 512 bit for the parameter of NIST category 1. CSIDH-ECIES-KEM is the most compact post-quantum CCA-secure KEM as far as we know.
- We discuss difficulty of extending ACE-KEM and FACE-KEM to CSIDH-based as CSIDH-PSEC-KEM and CSIDH-ECIES-KEM. Though the CSIDH system is compatible to classical DH, there is a gap between algebraic structures. Due to the gap it is impossible to extend these schemes with retaining original structures. Hence, it indicates that we do not always replace known DH-based constructions to CSIDH-based.

1.4 Related Work

Recently, several generic constructions of post-quantum CCA-secure KEM (i.e., in the QRO model) are studied. Boneh et al. [BODF+11] proved a KEM variant of Bellare-Rogaway construction based on a one-way trapdoor function is CCA-secure in the QRO model. Targhi and Unruh [TU16] proposed a variant of Fujisaki-Okamoto transformation and OAEP. Hofheinz et al. [HHK17] subsequently gave a modular analysis for the conversion. Saito et al. [SXY18] showed a construction of CCA-secure KEM based on a deterministic PKE scheme satisfying a new notion. Xagawa and Yamakawa [XY19] extended it to quantum CCA-security. These construction requires various properties for underlying primitives.

[1] Peikert [Pei19] showed a new quantum security analysis of CSIDH-512, corresponding to NIST category 1, by using the collimation sieve technique, and CSIDH-512 is broken by 40 bit quantum memory and 2^{16} quantum oracle queries (i.e., 56 bit quantum security). Hence, He estimates that the quantum security level of CSIDH-512 is rather weaker than NIST category 1. On the other hand, the quantum circuit for the group operation of CSIDH is very high cost. Thus, by considering such external overheads of circuits in addition to his evaluation, CSIDH-512 still seems safe in reality.

It is not clear that the CSIDH system satisfies such properties. Though some constructions [TU16,HHK17] only require CPA-security, the resultant CCA-secure KEM is less compact than our schemes. Also, Szepieniec et al. [SRP18] introduced a generic construction from noisy key agreement. We compare our schemes to KEM schemes obtained from the generic construction in Section 5.

2 Preliminaries

Throughout this paper we use the following notations. If S is a set, then by $s \in_R \mathsf{S}$ we denote that s is sampled uniformly from S. If $\mathcal{R}$ is an algorithm, then by $y \leftarrow \mathcal{R}(x; r)$ we denote that y is output by $\mathcal{R}$ on input x and randomness r (if $\mathcal{R}$ is deterministic, r is empty). The security parameter is κ.

2.1 Key Encapsulation Mechanism

Definition 1 (Syntax for KEM Schemes). *A KEM scheme Π consists of the following 3-tuple* (**Gen**, **Enc**, **Dec**)*:*

Gen : *a key generation algorithm which on input 1^κ, where κ is the security parameter, outputs a pair of keys (pk, sk).*

Enc : *an encapsulation algorithm which takes as input public key pk, outputs session key K and ciphertext CT.*

Dec : *a decapsulation algorithm which takes as input secret key sk and ciphertext CT, outputs session key K or reject symbol $\perp$.*

CCA-security is recognized as the strongest security notion. Here, we show the definition of CCA-security for KEM as follows.

Definition 2 (CCA-Security for KEM). *A KEM scheme Π is CCA-secure for KEM if the following property holds for security parameter κ; For any probabilistic polynomial-time (PPT) adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, the advantage* $\mathsf{Adv}^{\mathsf{kem}}_{\Pi}(\mathcal{A}) = |\Pr[(pk, sk) \leftarrow \mathbf{Gen}(1^\kappa);\ (state) \leftarrow \mathcal{A}_1^{\mathcal{DO}(sk,\cdot)}(pk);\ b \xleftarrow{R} \{0,1\};\ (K_0^*, CT_0^*) \leftarrow \mathbf{Enc}(pk);\ K_1^* \xleftarrow{R} \mathcal{K};\ b' \leftarrow \mathcal{A}_2^{\mathcal{DO}(sk,\cdot)}(pk,\ (K_b^*, CT_0^*),\ state);\ b' = b] - 1/2|$ *is negligible in κ, where $\mathcal{DO}$ is the decapsulation oracle which outputs K or $\perp$ on receiving CT, $\mathcal{K}$ is the space of session key, and state is state information which $\mathcal{A}$ wants to preserve from $\mathcal{A}_1$ to $\mathcal{A}_2$. $\mathcal{A}$ cannot submit the ciphertext $CT = CT_0^*$ to $\mathcal{DO}$.*

2.2 Hard Homogeneous Space and CSIDH

Here, we recall the definition of HHS [Cou06], and the CSIDH system [CLM+18] as an instantiation of HHS.

Hard Homogeneous Space. The definition of HHS is as follows.

Definition 3 (Freeness and Transitivity). *X denotes a finite set, and G denotes an abelian group. We say that G acts efficiently on X freely and transitively if there is an efficiently computable map $* : G \times X \to X$ as follows:*

- *for any $x \in X$ and $g, h \in G$, $g * (h * x) = (gh) * x$ holds, and there is an identity element $id \in G$ such that $id * x = x$,*
- *for any $(x, y) \in X \times X$, there is $g \in G$ such that $g * x = y$, and*
- *for any $x \in X$ and $g, h \in G$ such that $g * x = h * x$, $g = h$ holds.*

Definition 4 (Hard Homogeneous Space). *A HHS consists of a finite abelian group G acting freely and transitively on some set X such that the following tasks are efficiently executable:*

- *Computing the group operation on G*
- *Sampling randomly from G with (close to) uniform distribution*
- *Deciding validity and equality of a representation of elements of X*
- *Computing the action of a group element $g \in G$ on some $x \in X$ (i.e., $g * x$)*

CSIDH. The CSIDH system is an instantiation of HHS from $\mathbb{F}_p$-rational supersingular elliptic curves and their $\mathbb{F}_p$-rational isogeny. Let $\mathcal{E}\ell\ell_p(\mathcal{O})$ be the set of elliptic curves over $\mathbb{F}_p$ whose $\mathbb{F}_p$-rational endomorphism ring is some fixed quadratic order $\mathcal{O}$, and $\mathrm{cl}(\mathcal{O})$ be the ideal class group of $\mathcal{O}$. Then, the CSIDH system is regarded as HHS by setting $X = \mathcal{E}\ell\ell_p(\mathcal{O})$ and $G = \mathrm{cl}(\mathcal{O})$ as the parameter of HHS. For curve $E \in X$ and ideal class $[\mathfrak{g}] \in G$, the group action $[\mathfrak{g}] * E$ corresponds to the map $([\mathfrak{g}], E) \longmapsto E/\mathfrak{g}$. Since $E/\mathfrak{g}$ is a supersingular curve, the form of $E/\mathfrak{g}$ is $y^2 = x^3 + cx^2 + x$ for $c \in \mathbb{F}_p$. Then, $[\mathfrak{g}] * E$ can be represented as such Montgomery coefficient c.

Due to commutativity of $\mathrm{cl}(\mathcal{O})$, for $[\mathfrak{g}], [\mathfrak{g}'] \in G$, $E \in X$, $E_{\mathfrak{g}} = E/\mathfrak{g}$ and $E_{\mathfrak{g}'} = E/\mathfrak{g}'$, curves $E_{\mathfrak{g}'}/\mathfrak{g}$ and $E_{\mathfrak{g}}/\mathfrak{g}'$ are identical. Thus, we can use the Montgomery coefficient of $E/\mathfrak{g}\mathfrak{g}'$ (i.e., $([\mathfrak{g}][\mathfrak{g}']) * E$) as the common secret computed by two ways. Please see [CLM+18] for the detail of the mathematical foundation of the CSIDH system. In this paper, we use the notation of HHS as the CSIDH system for simplicity.

In the CSIDH system, hardness assumptions are defined as classical DH by using HHS. We recall the computational DH-type assumption for HHS defined in [BGK+18].[2]

Definition 5 (CSI-CDH Problem [BGK+18]). *For $E_0 \in X$, $[\mathfrak{a}], [\mathfrak{b}] \in_R G$, $E_{\mathfrak{a}} = [\mathfrak{a}] * E_0$ and $E_{\mathfrak{b}} = [\mathfrak{b}] * E_0$, the advantage of a PPT solver $\mathcal{S}$ in the CSI-CDH problem is defined as*

$$\mathsf{Adv}^{\mathsf{csi\text{-}cdh}}_{G,X}(\mathcal{S}) = \Pr \mathcal{S}(E_0, E_{\mathfrak{a}}, E_{\mathfrak{b}}) \to ([\mathfrak{a}][\mathfrak{b}]) * E_0.$$

[2] In [BGK+18], assumptions are defined as a generalized form for n-way by using cryptographic invariant maps (CIM). In the case of $n = 1$, CIM is the same as HHS.

Definition 6 (CSI-DDH Problem [BGK+18]). *For $E_0 \in X$, $[\mathfrak{a}], [\mathfrak{b}], [\mathfrak{c}] \in_R G$, $E_\mathfrak{a} = [\mathfrak{a}] * E_0$ and $E_\mathfrak{b} = [\mathfrak{b}] * E_0$, the advantage of a PPT distinguisher $\mathcal{D}$ in the CSI-DDH problem is defined as*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{csi\text{-}ddh}}_{G,X}(\mathcal{D}) = |\Pr[\mathcal{D}(E_0, E_\mathfrak{a}, E_\mathfrak{b}, E' = ([\mathfrak{a}][\mathfrak{b}]) * E_0) \to 1] \\ - \Pr[\mathcal{D}(E_0, E_\mathfrak{a}, E_\mathfrak{b}, E' = [\mathfrak{c}] * E_0) \to 1]|.\end{aligned}$$

Definition 7 (CSI-GDH Problem [FTY19]). *For $E_0 \in X$, $[\mathfrak{a}], [\mathfrak{b}] \in_R G$, $E_\mathfrak{a} = [\mathfrak{a}] * E_0$ and $E_\mathfrak{b} = [\mathfrak{b}] * E_0$, the advantage of a PPT solver $\mathcal{S}$ in the CSI-GDH problem is defined as*

$$\mathsf{Adv}^{\mathsf{csi\text{-}gdh}}_{G,X}(\mathcal{S}) = \Pr[\mathcal{S}^{\mathcal{DDH}(\cdot,\cdot,\cdot)}(E_0, E_\mathfrak{a}, E_\mathfrak{b}) \to ([\mathfrak{a}][\mathfrak{b}]) * E_0].$$

where $\mathcal{DDH}$ is the decision oracle which outputs 1 if the input $(E_{\mathfrak{a}'}, E_{\mathfrak{b}'}, E')$ satisfies $E_{\mathfrak{a}'} = [\mathfrak{a}'] * E_0, E_{\mathfrak{b}'} = [\mathfrak{b}'] * E_0$ and $E' = ([\mathfrak{a}'][\mathfrak{b}']) * E_0$, or outputs 0 otherwise.

The CSI-CDH (resp. CSI-DDH, CSI-GDH) problem corresponds to the classical computational DH (resp. decisional DH, gap DH) problem.[3]

Protocol of CSIDH. Here, we recall the protocol of CSIDH [CLM+18].

Public Parameters. Let $p = (4 \cdot \ell_1 \cdots \ell_{n-1})$ be a large prime where each ℓ_i is a small distinct odd prime. Then, the supersingular elliptic curve $E_0 : y^2 = x^3 + x$ over $\mathbb{F}_p$ with endomorphism ring $\mathcal{O} = \mathbb{Z}[\pi]$ is constructed where π is the Frobenius endomorphism satisfying $\pi^2 = -p$. For the notation of HHS, G is denoted by $\mathrm{cl}(\mathcal{O})$ and X is denoted by $\mathcal{E}\ell\ell_p(\mathcal{O})$; and thus, $E_0 \in X = \mathcal{E}\ell\ell_p(\mathcal{O})$. $[\mathfrak{g}] \in_R G$ means that integers $(e_1, \ldots, e_n)$ are randomly sampled from a range $\{-m, \ldots, m\}$ and $[\mathfrak{g}] = [\mathfrak{l}_1^{e_1} \cdots \mathfrak{l}_n^{e_n}] \in \mathrm{cl}(\mathcal{O})$ where $\mathfrak{l}_i = (\ell_i, \pi - 1)$. $[\mathfrak{g}] * E_0$ is represented by the Montgomery coefficient $c \in \mathbb{F}_p$ of the elliptic curve $[\mathfrak{g}]E_0 : y^2 = x^3 + cx^2 + x$ by applying the action of $[\mathfrak{g}]$ to E_0.

The public parameters are (G, X, E_0).

Session. Parties A and B executes a key exchange session as follows:

1. Party A chooses $[\mathfrak{a}] \in_R G$, and sends the public key $\hat{A} = [\mathfrak{a}] * E_0$ to party B.
2. Party B chooses $[\mathfrak{b}] \in_R G$, and sends the public key $\hat{B} = [\mathfrak{b}] * E_0$ to party A.
3. On receiving $\hat{B}$, party A generates the session key $SK = [\mathfrak{a}] * \hat{B}$.
4. On receiving $\hat{A}$, party B generates the session key $SK = [\mathfrak{b}] * \hat{A}$.

Since G is an abelian group, $[\mathfrak{a}][\mathfrak{b}] = [\mathfrak{b}][\mathfrak{a}]$ holds. Therefore, $[\mathfrak{a}] * \hat{B} = [\mathfrak{a}] * ([\mathfrak{b}] * E_0) = ([\mathfrak{a}][\mathfrak{b}]) * E_0 = ([\mathfrak{b}][\mathfrak{a}]) * E_0 = [\mathfrak{b}] * ([\mathfrak{a}] * E_0) = [\mathfrak{b}] * \hat{A}$ holds from Definition 3.

It is obvious that the session key SK is hard to find for any passive adversary if the CSI-CDH problem is hard.

[3] Dobson and Galbraith [DG19] show an attack to the gap DH-like assumption for SIDH [FTTY18]. The attack uses an algebraic structure of SIDH. Such an attack strategy is not applicable to CSIDH because of the difference between algebraic structures. Hence, the CSI-GDH assumption is considered to still seem safe.

Public Parameter: X, G, $E_0 \in X$, $H_1 : \{0,1\}^\kappa \to G \times \{0,1\}^\kappa$, $H_2 : X^2 \to \{0,1\}^\kappa$

$\mathbf{Gen}(1^\kappa)$	$\mathbf{Enc}(pk)$	$\mathbf{Dec}(sk, CT)$
$\mathfrak{s} \in_R G$	$t \in_R \{0,1\}^\kappa$	parse CT as (C_1, C_2)
$E_\mathfrak{s} = [\mathfrak{s}] * E_0$	$\mathfrak{r}\|\|K = H_1(t)$	$C' = [sk] * C_1$
$pk = E_\mathfrak{s}$	$C_1 = [\mathfrak{r}] * E_0$	$t = C_2 \oplus H_2(C_1, C')$
$sk = \mathfrak{s}$	$C' = [\mathfrak{r}] * pk$	$\mathfrak{r}\|\|K' = H_1(t)$
return (pk, sk)	$C_2 = t \oplus H_2(C_1, C')$	if $C_1 \neq [\mathfrak{r}] * E_0$, $K = \perp$
	$CT = (C_1, C_2)$	else $K = K'$
	return (CT, K)	**return** K

Fig. 1: CSIDH-PSEC-KEM

3 CSIDH-PSEC-KEM

PSEC is the abbreviation of "Provably Secure Elliptic Curve encryption". It is a DH-based PKE scheme developed at Nippon Telegraph and Telephone corporation [NTT08] based on the work of Fujisaki and Okamoto [FO99]. PSEC-KEM is the KEM version of PSEC, which is standardized in ISO/IEC 18033-2 [ISO]. In the draft [Sho01] of ISO/IEC 18033-2, CCA-security of PSEC-KEM is proved under the CDH assumption in the RO model.

CSIDH-PSEC-KEM is a natural extension of PSEC-KEM to CSIDH-based with retaining the structure. Instead of the scalar multiplication on the elliptic curve, we use the group action operation $*$ to generate the public key and the ciphertext. The protocol of CSIDH-PSEC-KEM is given in Fig. 1. X and G are parameters of HHS. $H_1 : \{0,1\}^\kappa \to G \times \{0,1\}^\kappa$ and $H_2 : X^2 \to \{0,1\}^\kappa$ are hash functions modeled as ROs.

The first advantage of CSIDH-PSEC-KEM against SIKE-KEM [SIKE17] is the ciphertext overhead. The ciphertext of SIKE-KEM contains an element of the SIDH public key and a κ-bit element. On the other hand, the ciphertext of CSIDH-PSEC-KEM contains an element of the CSIDH public key and a κ-bit element. Since it is estimated that the CSIDH public key is more compact than the SIDH public key for equivalent security level [SIKE17,CLM+18], CSIDH-PSEC-KEM is more efficient in the ciphertext overhead, than SIKE-KEM.

The second advantage is the public key size. The public key of SIKE-KEM contains an element of the SIDH public key. The public key of CSIDH-PSEC-KEM contains an element of the CSIDH public key. Thanks to compactness of the CSIDH public key, CSIDH-PSEC-KEM is also more efficient in the public key size, than SIKE-KEM.

The third advantage is security in the QRO model. Naturally, by a similar manner as the security proof of the original PSEC-KEM, we can prove CCA-security of CSIDH-PSEC-KEM under the CSI-CDH assumption in the classical RO model. In this paper, we give a security proof in the QRO model. In our proof, the CSI-DDH assumption is necessary instead of the CIS-CDH assumption. The reason that it is not easy to prove the security under the CSI-CDH assumption

in the QRO model is as follows; In the reduction to the CCA-security, the CSI-CDH solver needs to extract the answer of the CSI-CDH problem from a hash query by the CCA adversary. However, the query may be a quantum state (i.e., superposition), and the solver cannot record a copy of the input due to the no-cloning theorem. Thus, such a proof strategy (as the original PSEC-KEM) does not work. Conversely, the CSI-DDH just distinguishes if the given instance is valid, and does not need extract hash queries. Hence, we can prove CCA-security under the CSI-DDH assumption by using some proof techniques (see Section 3.1).

Though the computational time of CSIDH is slower than SIDH, it is practical enough as estimated in [CLM+18]. A detailed comparison to SIKE-KEM is shown in Section 5.

3.1 Useful Techniques for Quantum Random Oracle Model

A hurdle on security proofs in the quantum random oracle model is how to generate random values for exponentially many positions in order to simulate outputs of the hash function. For a hash function $H : Dom \to Rng$, in the quantum random oracle model, the adversary poses a superposition $|\phi\rangle = \Sigma\alpha_x|x\rangle$ and the oracle returns $\Sigma\alpha_x|H(x)\rangle$. If Rng is large for a quantum polynomial-time simulator, it is difficult to generate all random output values of H to compute $\Sigma\alpha_x|H(x)\rangle$. Zhandry [Zha12] showed a solution with the notion of k-wise independent function.

A weight assignment on a set $\mathcal{X}$ is a function $D : \mathcal{X} \to \mathbb{R}$ such that $\Sigma_{x\in\mathcal{X}} D(x) = 1$. A distribution on $\mathcal{X}$ is a weight-assignment D such that $D(x) \geq 0$ for all $x \in \mathcal{X}$. Consider the set of functions $H : \mathcal{X} \to \mathcal{Y}$ for sets $\mathcal{X}$ and $\mathcal{Y}$, denoted by $H_{\mathcal{X},\mathcal{Y}}$. We define the marginal weight assignment $D_{\mathcal{W}}$ of D on $H_{\mathcal{X},\mathcal{Y}}$ where the weight of a function $H_{\mathcal{W}} : \mathcal{W} \to \mathcal{Y}$ is equal to the sum of the weights of all $H \in H_{\mathcal{X},\mathcal{Y}}$ that agree with $H_{\mathcal{W}}$ on $\mathcal{W}$.

Definition 8 (k-wise equivalence). *We call two weight assignments D_1 and D_2 on $H_{\mathcal{X},\mathcal{Y}}$ k-wise equivalent if for all $\mathcal{W} \subseteq \mathcal{X}$ of size k, the marginal weight assignments $D_{1,\mathcal{W}}$ and $D_{2,\mathcal{W}}$ (of D_1 and D_2) over $H_{\mathcal{X},\mathcal{Y}}$ are identical.*

Definition 9 (k-wise independent function). *We call a function f k-wise independent function if f is k-wise equivalent to a random function.*

Lemma 1 (Theorem 3.1 in [Zha12]). *Let A be a quantum algorithm making q quantum queries to an oracle $H : \mathcal{X} \to \mathcal{Y}$. If we draw H from some weight assignment D, then for every z, the quantity $\Pr_{H\leftarrow D}[A^H() = z]$ is a linear combination of the quantities $\Pr_{H\leftarrow D}[H(x_i) = r_i \ \forall i \in 1, \ldots, 2q]$ for all possible settings of the x_i and r_i.*

Lemma 2 (Theorem 6.1 in [Zha12]). *If there exists $2q_i$-wise independent function, then any quantum algorithm A making q_i quantum queries to random oracles O_i can be efficiently simulated by a quantum algorithm B, which has the same output distribution, but makes no queries.*

Hence, a quantum algorithm B can simulate quantum random oracles in a polynomial-time. We use this simulation technique to simulate outputs of hash functions H_1 and H_2 in the security proof of CSIDH-PSEC-KEM.

On the other hand, the other problem on security proofs in the quantum random oracle model is how to insert intended random values as the outputs of corresponding oracle inputs. Zhandry [Zha12] showed a solution with the notion of semi-constant distributions $\mathbf{SC}_\omega$.

Definition 10 (Semi-constant distribution). *We define $\mathbf{SC}_\omega$, the semi-constant distribution, as the distribution over $H_{\mathcal{X},\mathcal{Y}}$ resulting from the following process:*

- *First, pick a random element y from $\mathcal{Y}$.*
- *For each $x \in \mathcal{X}$, do one of the following:*
  - *With probability ω, set $H(x) = y$. We call x a distinguished input to H.*
  - *Otherwise, set $H(x)$ to be a random element in $\mathcal{Y}$.*

Lemma 3 (Corollary 4.3 in [Zha12]). *The distribution of outputs of a quantum algorithm making q_H queries to an oracle drawn from $\mathbf{SC}_\omega$ is at most a distance $\frac{3}{8}q_H^4\omega^2$ away from the case when the oracle is drawn from the uniform distribution.*

We suppose that the simulation succeeds with probability ϵ if the adversary uses an inserted random value as the outputs of corresponding oracle inputs. If the probability that the adversary uses one of the points is ω, then the simulation succeeds with probability $\epsilon\omega - \frac{3}{8}q_H^4\omega^2$. By choosing ω to maximize the success probability, the simulation succeeds with probability $O(\epsilon^2/q_H^4)$. We use this simulation technique to insert a CSI-DDH instance into an output of the hash function H_2 in the security proof of CSIDH-PSEC-KEM.

3.2 Security

We show that CSIDH-PSEC-KEM is CCA-secure under the CSI-DDH assumption in the QRO model.

Theorem 1 (Security of CSIDH-PSEC-KEM). *For the advantage $\mathsf{Adv}^{\mathsf{csi\text{-}ddh}}_{G,X}(\mathcal{D})$ of the CSI-DDH problem, the advantage $\mathsf{Adv}^{\mathsf{kem}}_{\mathsf{psec}}(\mathcal{A})$ of CSIDH-PSEC-KEM is as follows in the QRO model:*

$$\mathsf{Adv}^{\mathsf{kem}}_{\mathsf{psec}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{csi\text{-}ddh}}_{G,X}(\mathcal{D})^{1/2}\left(\frac{3}{4}(q_{H_2}+q_D+1)^4 + 2q_{H_2}q_D\right)^{1/2} + \frac{q_{H_1}+2q_D}{2^\kappa}.$$

where q_{H_1}, q_{H_2} and q_D denote the upper bound of queries to H_1, H_2 and $\mathcal{DO}$, respectively.

First, we give an intuition of the proof.

We use a game hopping technique. First, we change the rules of the decapsulation oracle $\mathcal{DO}$ such that the secret key is not used. Such game transitions are indistinguishable by the property of ROs. Next, we change the construction

of the challenge ciphertext and session key such that the session key is a fixed value ζ as the output of H_2. Then, we show that the advantage of the KEM adversary $\mathcal{A}$ is equivalent to the advantage of the CSI-DDH distinguisher $\mathcal{D}$; that is, we construct $\mathcal{D}$ (the input is $(E_0, E_{\mathfrak{a}}, E_{\mathfrak{b}}, E')$) from $\mathcal{A}$. In the final game, the advantage of $\mathcal{A}$ is negligible.

$\mathcal{D}$ has difficulty in responding to hash queries because it needs to return superpositions corresponding to random values for exponentially many positions (the domain of H_1 is $\{0,1\}^\kappa$ and the domain of H_2 is X^2). We solve this problem by using Lemma 2. Specifically, since the number of queries to H_1 (resp. H_2) made by $\mathcal{A}$ is q_{H_1} (resp. q_{H_2}) for direct queries, q_D for decapsulation queries, and one for the challenge ciphertext, for the total of $q_{H_1}+q_D+1$ (resp. $q_{H_2}+q_D+1$) queries, a $(q_{H_1}+q_D+1)$-wise independent function (resp. a $(q_{H_2}+q_D+1)$-wise independent function) is sufficient to simulate superposition of outputs.

There is the other difficulty to correctly answer the CSI-DDH problem. If the position of ζ is only the corresponding input in the superposition, $\mathcal{A}$ uses ζ to distinguish the real session key from a random session key with exponentially small probability. We can also solve this problem by using Lemma 3. Specifically, $\mathcal{D}$ inserts ζ in outputs for inputs $\mathcal{X} \subset X^2$. The probability that a randomly chosen input is contained in $\mathcal{X}$ is ω. Then, H_2 is distributed according to $\mathbf{SC}_\omega$, and $\mathcal{A}$ can only tell it is not random with probability $O(\mathsf{Adv}^{\mathsf{kem}}_{\mathsf{psec}}(\mathcal{A})^2/(q_{H_2}+q_D+1)^4)$ from Lemma 3.

Therefore, $\mathcal{D}$ can use the distinguishing capacity of $\mathcal{A}$ to distinguish the CSI-DDH challenge.

Proof. We change the interface of the decapsulation oracle query and the computation of the challenge ciphertext and session key. These instances are gradually changed over seven hybrid experiments, depending on specific sub-cases. We denote these hybrid experiments by $\mathbf{H}_0, \ldots, \mathbf{H}_6$ and the advantage of the adversary $\mathcal{A}$ when participating in experiment $\mathbf{H}_i$ by $\mathsf{Adv}(\mathcal{A}, \mathbf{H}_i)$.

Hybrid experiment $\mathbf{H}_0$: This experiment denotes the real experiment for CCA-security and in this experiment the decapsulation oracle $\mathcal{DO}$ is as defined in the protocol. (C_1^*, C_2^*) and K^* denote the challenge ciphertext and the session key, respectively. Also, let $\rho(t) = \mathfrak{r}$ such that $\mathfrak{r}||K = H_1(t)$, and t^* be randomness to be used to generate the challenge ciphertext and the session key. Thus, $\mathsf{Adv}(\mathcal{A}, \mathbf{H}_0)$ is the same as the advantage of the real experiment.

Hybrid experiment $\mathbf{H}_1$: In this experiment, the rule of $\mathcal{DO}$ is changed as follows; If $(C_1 = C_1^*, C_2)$ is posed, the query is rejected.

Let F_1 be the event that in $\mathbf{H}_1$ such a ciphertext is rejected that would not have been rejected under the rules of $\mathbf{H}_0$. From the deference lemma [Sho04], we have $|\mathsf{Adv}(\mathcal{A}, \mathbf{H}_1) - \mathsf{Adv}(\mathcal{A}, \mathbf{H}_0)| \leq Pr[F_1]$. Since $\mathcal{A}$ cannot pose the challenge ciphertext to $\mathcal{DO}$, $C_2 \neq C_2^*$ holds. Hence, t for C_2 must be different from t^* because $C_2 = C_2^*$ if $t = t^*$. Also, since (C_1^*, C_2) is not rejected in $\mathbf{H}_0$, $\mathcal{A}$ must

find $\mathfrak{r} = \rho(t) = \rho(t^*)$. It means the probability of collision; and thus, $\Pr[F_1] \leq (q_{H_1} + q_D)/2^\kappa$.

Therefore, we obtain

$$|\mathsf{Adv}(\mathcal{A}, \mathbf{H}_1) - \mathsf{Adv}(\mathcal{A}, \mathbf{H}_0)| \leq (q_{H_1} + q_D)/2^\kappa.$$

Hybrid experiment $\mathbf{H}_2$: In this experiment, the rule of $\mathcal{DO}$ is changed again as follows; If $(C_1 \neq C_1^*, C_2)$ is posed and t is never posed to H_1, the query is rejected.

Let F_2 be the event that in $\mathbf{H}_2$ such a ciphertext is rejected that would not have been rejected under the rules of $\mathbf{H}_1$. First, we consider the case of $t = t^*$. In this case, since $C_1 \neq C_1^* = [\rho(t)] * E_0$ holds, (C_1, C_2) is always rejected by the decapsulation procedure in $\mathbf{H}_1$. Next, we consider the case of $t \neq t^*$. In this case, since t is never posed to H_1, $\rho(t)$ is perfectly random from $\mathcal{A}$'s view. Hence, the probability that $C_1 = [\rho(t)] * E_0$ is $q_D/2^\kappa$. Therefore, we obtain

$$|\mathsf{Adv}(\mathcal{A}, \mathbf{H}_2) - \mathsf{Adv}(\mathcal{A}, \mathbf{H}_1)| \leq q_D/2^\kappa.$$

Hybrid experiment $\mathbf{H}_3$: In this experiment, the rule of $\mathcal{DO}$ is changed again as follows; If $(C_1 \neq C_1^*, C_2)$ is posed, and $C_1 \neq [\rho(t')] * E_0$ for any t' posed to H_1, then the query is rejected. Also, if $(C_1 \neq C_1^*, C_2)$ is posed, and there exists t' such that $C_1 = [\rho(t')] * E_0$ and t' is posed to H_1, then C' in the decapsulation procedure is computed by $[\rho(t')] * E_{\mathfrak{b}}$ instead of $[\mathfrak{s}] * C_1$. It means that $\mathcal{DO}$ does not use the secret key.

First, we consider the case that $C_1 \neq [\rho(t')] * E_0$ for any t' posed to H_1. For t' which is not posed to H_1, the query is rejected by the rule of $\mathbf{H}_2$. For t' which is posed to H_1, the query is also rejected in $\mathbf{H}_2$ because $C_1 \neq [\rho(t')] * E_0$ in the decapsulation procedure. Hence, there is no difference between rejected queries in $\mathbf{H}_2$ and $\mathbf{H}_3$. Next, we consider the case that there exists t' such that $C_1 = [\rho(t')] * E_0$ and t' is posed to H_1. There is no difference between rejected queries in $\mathbf{H}_2$ and $\mathbf{H}_3$ because $[\rho(t')] * E_{\mathfrak{b}} = [\mathfrak{s}] * C_1$ always holds in this case. Therefore, we obtain

$$\mathsf{Adv}(\mathcal{A}, \mathbf{H}_3) = \mathsf{Adv}(\mathcal{A}, \mathbf{H}_2).$$

Hybrid experiment $\mathbf{H}_4$: In this experiment, the rule of halting is changed as follows; Let $\omega \in (0, 1)$ be chosen later, and $\mathcal{X}$ be a subset of X^2 where $(\hat{C}_1, \hat{C}') \in_R X^2$ is put in $\mathcal{X}$ with independent probability ω. $\mathbf{H}_4$ halts if $(E_{\mathfrak{a}}, E') \notin \mathcal{X}$ (where $(E_{\mathfrak{a}}, E')$ is a part of the CSI-DDH instance), $\mathcal{A}$ poses $H_2(C_1, C')$ such that $(C_1, C') \in \mathcal{X}$, or $\mathcal{A}$ poses $\mathcal{DO}(C_1, C_2)$ such that $(C_1, C') \in \mathcal{X}$.

We obtain

$$\begin{aligned}\mathsf{Adv}(\mathcal{A}, \mathbf{H}_4) &\geq \omega(1 - \omega q_{H_2} q_D) \cdot \mathsf{Adv}(\mathcal{A}, \mathbf{H}_3) \\ &\geq \omega \mathsf{Adv}(\mathcal{A}, \mathbf{H}_3) - \omega^2 q_{H_2} q_D.\end{aligned}$$

Hybrid experiment $\mathbf{H}_5$: In this experiment, the rule of H_2 is changed as follows; ζ is set as $H_2(C_1, C')$ for all $(C_1, C') \in \mathcal{X}$, and hash values are randomly chosen for all other inputs. Now, H_2 is distributed according to $\mathbf{SC}_\omega$. By Lemma 3, the output distribution of $\mathcal{A}$ in $\mathbf{H}_5$ is at most a distance $\frac{3}{8}(q_{H_2} + q_D + 1)^4\omega^2$ from that in $\mathbf{H}_4$.

Therefore, we obtain

$$\mathsf{Adv}(\mathcal{A}, \mathbf{H}_5) \geq \mathsf{Adv}(\mathcal{A}, \mathbf{H}_4) - \frac{3}{8}(q_{H_2} + q_D + 1)^4\omega^2.$$

Hybrid experiment $\mathbf{H}_6$: In this experiment, the rule of generating the challenge ciphertext and session key is changed as follows; $\mathfrak{r}^* \in_R G$ and $K^* \in_R \{0,1\}^\kappa$ are randomly chosen instead of computing $H_1(t^*)$.

We construct a distinguisher $\mathcal{D}$ of the CSI-DDH problem with the advantage $\mathsf{Adv}^{\mathsf{csi\text{-}ddh}}_{G,X}(\mathcal{D})$ from $\mathcal{A}$. For simplicity, we assume that $\mathcal{D}$ has quantum access to three random oracles $\hat{H}_1 : \{0,1\}^\kappa \to G \times \{0,1\}^\kappa$, $\hat{H}_2 : X^2 \to \{0,1\}^\kappa$ and $\hat{H}'_2 : X^2 \to \{0,1\}$ where $\hat{H}'_2$ outputs 1 with probability ω. Let $\mathcal{X}$ be the set of (C_1, C') such that $H_2(C_1, C') = 1$. We can see that the above conditions are equivalent to $\mathbf{H}_6$. By Lemma 2, $\mathcal{D}$ can perfectly simulate $\hat{H}_1$ and $(\hat{H}_2, \hat{H}'_2)$ by using a $(q_{H_1} + q_D + 1)$-wise independent function and a $(q_{H_2} + q_D + 1)$-wise independent function without oracle accesses, respectively. $\mathcal{L}$ is a list which is initially empty and maintained by $\mathcal{D}$.

- **Input and Setting of Public Key.** $\mathcal{D}$ receives the challenge instance $(E_0, E_\mathfrak{a}, E_\mathfrak{b}, E')$. Then, $\mathcal{D}$ sends the public parameter E_0, and $pk = E_\mathfrak{b}$ to $\mathcal{A}$.

- **Simulation of Challenge Ciphertext and Session key.** $\mathcal{D}$ chooses $t^* \in_R \{0,1\}^\kappa$ and $\zeta \in_R \{0,1\}^\kappa$, and sets $CT^* = (E_\mathfrak{a}, t^* \oplus \zeta)$ as the challenge ciphertext. Also, $\mathcal{D}$ computes K^* as $\mathbf{H}_5$.

- **Simulation of H_1.** On receiving $\hat{t}$, $\mathcal{D}$ simulates H_1 such that $H_1(\hat{t}) = \hat{H}_1(\hat{t})$.

- **Simulation of H_2.** On receiving $(\hat{C}_1, \hat{C}')$, $\mathcal{D}$ simulates H_2 such that

$$H_2(\hat{C}_1, \hat{C}') = \begin{cases} \zeta & \text{if } \hat{H}'_2(\hat{C}_1, \hat{C}') = 1 \\ \hat{H}_2(\hat{C}_1, \hat{C}') & \text{otherwise} \end{cases}$$

- **Simulation of $\mathcal{DO}$.** On receiving $(\hat{C}_1, \hat{C}_2)$, $\mathcal{D}$ simulates $\mathcal{DO}$ as $\mathbf{H}_3$ without using the secret key.

- **Analysis of Success Probability.** If $\mathcal{A}$ poses queries included in $\mathcal{X}$ to H_2 or $\mathcal{DO}$, then $\mathcal{A}$ distinguishes the simulation from the real experiment. However, in $\mathbf{H}_6$, these events do not occur because of the game hopping in $\mathbf{H}_4$. Also, $(E_\mathfrak{a}, E') \in \mathcal{X}$ holds. In the case of $E' = ([\mathfrak{a}][\mathfrak{b}]) * E_0$, the simulation

of the challenge ciphertext and session key is the same as $\mathbf{H}_5$. Then, $\mathcal{A}$ poses $(E_{\mathfrak{a}}, E')$ to H_2 and succeeds with $\mathsf{Adv}(\mathcal{A}, \mathbf{H}_5)$. In the case of $z = [\mathfrak{c}] * E_0$, the simulated challenge ciphertext and session key are independent because $H_2(E_{\mathfrak{a}}, E')$ is randomly chosen. Then, $\mathcal{A}$ succeeds with $\mathsf{Adv}(\mathcal{A}, \mathbf{H}_6)$. Therefore, we obtain

$$|\mathsf{Adv}(\mathcal{A}, \mathbf{H}_6) - \mathsf{Adv}(\mathcal{A}, \mathbf{H}_5)| \leq \mathsf{Adv}^{\mathsf{csi\text{-}ddh}}_{G,X}(\mathcal{D}).$$

Analysis of $\mathsf{Adv}(\mathcal{A}, \mathbf{H}_6)$: The hidden bit b is independent of the challenge session key. Therefore, we obtain

$$\mathsf{Adv}(\mathcal{A}, \mathbf{H}_6) = 0.$$

Then, by combining advantages we obtain

$$\mathsf{Adv}^{\mathsf{kem}}_{\mathsf{psec}}(\mathcal{A}) \leq \frac{1}{\omega}\mathsf{Adv}^{\mathsf{csi\text{-}ddh}}_{G,X}(\mathcal{D}) + \omega\left(\frac{3}{8}(q_{H_2} + q_D + 1)^4 + q_{H_2}q_D\right) + \frac{q_{H_1} + 2q_D}{2^\kappa}.$$

Since the right side is minimized when $\omega = \left(\frac{\mathsf{Adv}^{\mathsf{csi\text{-}ddh}}_{G,X}(\mathcal{D})}{\frac{3}{8}(q_{H_2}+q_D+1)^4+q_{H_2}q_D}\right)^{1/2}$, we obtain

$$\mathsf{Adv}^{\mathsf{kem}}_{\mathsf{psec}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{csi\text{-}ddh}}_{G,X}(\mathcal{D})^{1/2}\left(\frac{3}{4}(q_{H_2} + q_D + 1)^4 + 2q_{H_2}q_D\right)^{1/2} + \frac{q_{H_1} + 2q_D}{2^\kappa}.$$

□

4 CSIDH-ECIES-KEM

ECIES is the abbreviation of "Elliptic Curve Integrated Encryption Scheme". It is a DH-based hybrid encryption scheme based on DHAES [ABR99]. ECIES-KEM is the KEM version of ECIES, which is standardised in ISO/IEC 18033-2 [ISO]. It is a hashed variant of ElGamal KEM. In the draft [Sho01] of ISO/IEC 18033-2, CCA-security of ECIES-KEM is proved under the gap DH assumption in the RO model.

CSIDH-ECIES-KEM is a natural extension of ECIES-KEM to CSIDH-based with retaining the structure. The protocol of CSIDH-ECIES-KEM is given in Fig. 2. X and G are parameters of HHS. $H : X^2 \to \{0,1\}^\kappa$ is a hash function modeled as an RO.

CSIDH-ECIES-KEM contains the validation of the ciphertext in decapsulation (i.e., checking $CT \in X$) in order to ensure CCA-security by preventing active attacks such as Galbraith et al.'s attack [GPST16]. The validation can be efficiently done thanks to the CSIDH system. However, it is costly in the SIDH system because the SIDH system does not have an efficient validation method yet and countermeasures are expensive as mentioned in [GPST16].

The main advantage of CSIDH-ECIES-KEM is that the ciphertext is very compact. The ciphertext of CSIDH-ECIES-KEM only contains an element of

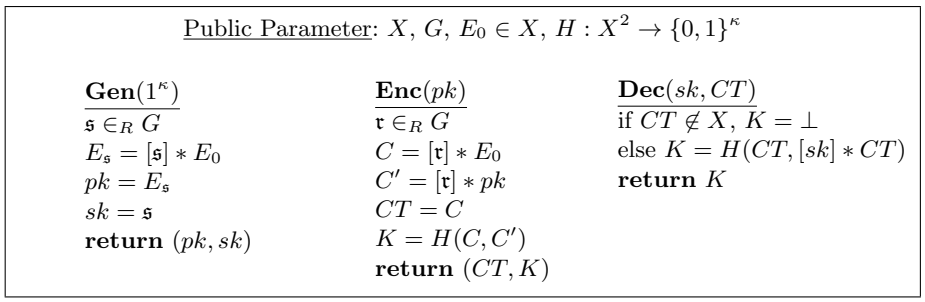

Fig. 2: CSIDH-ECIES-KEM

the CSIDH public key. The ciphertext overhead of CSIDH-ECIES-KEM is 512 bit for the parameter corresponding to NIST category 1 [NIST] (i.e., 128 bit security). It is just twice as much as the elliptic curve ElGamal cryptosystem.

Moreover, the other advantage is the security reduction is equivalent to the CSI-GDH assumption as shown in 2. Since SIKE-KEM uses the generic conversion [HHK17], the reduction is not tight. The reduction of CSIDH-PSEC-KEM is also not tight due to proving security in the QRO model.

It is not easy to prove security in the QRO model by a similar reason described in Section 3. The solver of the CSI-GDH problem must extract the answer of the problem from a hash query by the CCA adversary, but the solver cannot record a copy of the input due to the no-cloning theorem.

4.1 Security

We show that CSIDH-ECIES-KEM is CCA-secure under the CSI-GDH assumption in the RO model.

Theorem 2 (Security of CSIDH-ECIES-KEM). *For the advantage* $\mathsf{Adv}^{\mathsf{csi\text{-}gdh}}_{G,X}(\mathcal{S})$ *of the CSI-GDH problem, the advantage* $\mathsf{Adv}^{\mathsf{kem}}_{\mathsf{ecies}}(\mathcal{A})$ *of CSIDH-ECIES-KEM is as follows in the RO model:*

$$\mathsf{Adv}^{\mathsf{kem}}_{\mathsf{ecies}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{csi\text{-}gdh}}_{G,X}(\mathcal{S}).$$

Proof. We construct a solver $\mathcal{S}$ of the CSI-GDH problem with the advantage $\mathsf{Adv}^{\mathsf{csi\text{-}gdh}}_{G,X}(\mathcal{S})$ by assuming that there exists an adversary $\mathcal{A}$ of CSIDH-ECIES-KEM with the advantage $\mathsf{Adv}^{\mathsf{kem}}_{\mathsf{ecies}}(\mathcal{A})$. $\mathcal{L}$ is a list which is initially empty and maintained by $\mathcal{S}$.

- **Input and Setting of Public Key.** $\mathcal{S}$ receives the challenge instance $(E_0, E_\mathfrak{a}, E_\mathfrak{b})$. Then, $\mathcal{S}$ sends the public parameter E_0, and $pk = E_\mathfrak{b}$ to $\mathcal{A}$.

- **Simulation of Challenge Ciphertext and Session key.** $\mathcal{S}$ sets $CT^* = E_\mathfrak{a}$ as the challenge ciphertext. Also, $\mathcal{S}$ generates $K^* \in_R \{0,1\}^\kappa$ as the challenge session key.

Table 1: Comparison among CCA-secure KEM from isogeny

	Model	Assumption	Public key size	Ciphertext overhead	Time per encapsulation	Time per decapsulation
SIKE-KEM [SIKE17]	ROM	SI-CDH	2640 bit	3152 bit	≈ 3.1 ms	≈ 3.3 ms
SIDH-SRP [SRP18]	QROM	SI-DDH	2640 bit	3280 bit	≈ 3.1 ms	≈ 6.4 ms
CSIDH-SRP [SRP18]	QROM	CSI-DDH	512 bit	768 bit	≈ 81.6 ms	≈ 122.4 ms
CSIDH-PSEC-KEM	QROM	CSI-DDH	512 bit	640 bit	≈ 81.6 ms	≈ 81.6 ms
CSIDH-ECIES-KEM	ROM	CSI-GDH	512 bit	512 bit	≈ 81.6 ms	≈ 42.9 ms

– **Simulation of H.** On receiving $(\hat{C}, \hat{C}')$, $\mathcal{S}$ simulates H as follows:
  - If $(\hat{C}, \hat{C}', \hat{K}) \in \mathcal{L}$ for some $\hat{K} \in \{0,1\}^\kappa$, then return $\hat{K}$.
  - Else if $\mathcal{DDH}(E_\mathfrak{b}, \hat{C}, \hat{C}') = 1$ and $\hat{C} = E_\mathfrak{a}$, then outputs $\hat{C}'$ as the answer of the CSI-GDH problem.
  - Else if $\mathcal{DDH}(E_\mathfrak{b}, \hat{C}, \hat{C}') = 1$ and $(\hat{C}, \perp, \hat{K}) \in \mathcal{L}$ for some $\hat{K} \in \{0,1\}^\kappa$, then return $\hat{K}$ and store $(\hat{C}, \hat{C}', \hat{K})$ to $\mathcal{L}$.
  - Otherwise, generate $\hat{K} \in_R \{0,1\}^\kappa$, return $\hat{K}$ and store $(\hat{C}, \hat{C}', \hat{K})$ to $\mathcal{L}$.

– **Simulation of $\mathcal{DO}$.** On receiving $\hat{C}$, $\mathcal{S}$ simulates $\mathcal{DO}$ as follows:
  - If $\hat{C} \notin X$, then return $\perp$.
  - Else if $(\hat{C}, \hat{C}', \hat{K}) \in \mathcal{L}$ for some $\hat{C}' \in X$ and $\hat{K} \in \{0,1\}^\kappa$, then return $\hat{K}$.
  - Else if $(\hat{C}, \perp, \hat{K}) \in \mathcal{L}$ for some $\hat{K} \in \{0,1\}^\kappa$, then return $\hat{K}$.
  - Otherwise, generate $\hat{K} \in_R \{0,1\}^\kappa$, return $\hat{K}$ and store $(\hat{C}, \perp, \hat{K})$ to $\mathcal{L}$.

– **Analysis of Success Probability.** The simulation fails if $\mathcal{A}$ distinguishes the simulated challenge session key (i.e., chosen randomly) from the real challenge one (i.e, generated by $H(E_\mathfrak{a}, ([\mathfrak{a}][\mathfrak{b}]) * E_0)$). However, since H is an RO, $\mathcal{A}$ cannot obtain any information of $H(E_\mathfrak{a}, ([\mathfrak{a}][\mathfrak{b}]) * E_0)$ unless $\mathcal{A}$ poses $(E_\mathfrak{a}, ([\mathfrak{a}][\mathfrak{b}]) * E_0)$ to the RO. Hence, $\mathcal{A}$ must pose $(E_\mathfrak{a}, ([\mathfrak{a}][\mathfrak{b}]) * E_0)$ and obtain the answer of the CSI-GDH problem to H in order to distinguish these. When $\mathcal{A}$ poses it, $\mathcal{S}$ wins by the simulation of H. Other simulations are obviously perfect. Therefore, we obtain

$$\mathsf{Adv}^{\mathsf{kem}}_{\mathsf{ecies}}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{csi\text{-}gdh}}_{G,X}(\mathcal{S}).$$

□

5 Comparison

In this section, we give an comparison efficiency of our schemes and previous isogeny-based CCA-secure KEM schemes. The comparison is shown in Table 1.

To compare SIDH-based schemes and CSIDH-based schemes, we use parameters having the same security level (i.e., NIST category 1 [NIST]) corresponding to the key search on a block cipher with a 128 bit key (i.e., $\kappa = 128$). For SIDH, the parameter corresponding to NIST category 1 is estimated as SIKE$p434$ in [SIKE17]. As computational time of encapsulation and decapsulation of SIKE-KEM, we use the performance evaluation of x64-assembly implementation on

a 3.4GHz Intel Core i7-6700 (Skylake) processor in [SIKE17, Table 2.1]. The SI-CDH (resp. SI-DDH) assumption denotes the CDH-like (resp. DDH-like) assumption corresponding to SIDH. For CSIDH, the parameter corresponding to NIST category 1 is estimated as CSIDH-512 in [CLM+18]. The public key size and the ciphertext overhead are estimated as 512 bit. Computational time of a group action and time for a public key validation are about 40.3 ms and about 1.6 ms, respectively, based on the proof-of-concept implementation on a 3.5GHz Intel Core i5 (Skylake) processor in [CLM+18, Table 2]. CSIDH-PSEC-KEM contains a CSIDH public key as the public key, a CSIDH public key and a κ bit string as the ciphertext, and two group actions both for encapsulation and decapsulation. Also, CSIDH-ECIES-KEM contains a CSIDH public key as the public key, a CSIDH public key as the ciphertext, two group actions for encapsulation, and a group action and a ciphertext validation for decapsulation. We simply add these values without any acceleration technique.

Note that SIKE$p434$ is rather conservatively estimated to be at NIST category 1, whereas the estimates for CSIDH-512 require much stronger assumptions on the real-world attack costs and it is heavily debated if it achieves this security level. Also, since the best attacks against SIKE are exponential but the best attacks against CSIDH are subexponential, CSIDH becomes much more inefficient than SIKE at higher security levels. Hence, our comparison is not rigorous, and the values are just a guide for readers.

In the comparison, we also compare our schemes to KEM schemes using a generic construction [SRP18] from noisy key agreement. By using SIDH (or CSIDH) a CCA-secure KEM scheme in the QRO model is obtained. The resultant KEM scheme contains a SIDH (or CSIDH) public key as the public key, a SIDH (CSIDH) public key and two κ bit strings as the ciphertext, two group actions for encapsulation, and three group actions for decapsulation. Since sesion key indistinguishability is required for noisy key agreement, the SI-DDH assumption (or the CSI-DDH assumption) is necessary.

As shown in Table 1, CSIDH-PSEC-KEM is the most compact scheme which is secure in the QRO model, and CSIDH-ECIES-KEM is the most compact scheme compared to other schemes. Since SIKE-KEM is the most compact scheme in submissions to the NIST PQC competition, our schemes are also more compact than all submitted cryptosystems. The disadvantage of our schemes is computation time. However, both times for encapsulation and decapsulation are faster than 100 ms; and hence, it is still practical.

6 How about ACE-KEM and FACE-KEM?

ISO/IEC 18033-2 [ISO] contains other DH-based KEM schemes, ACE-KEM and FACE-KEM. ACE-KEM is based on Cramer-Shoup cryptosystem [CS98,CS04] and FACE-KEM is based on Kurosawa-Phong cryptosystem [KP14]. These schemes are proved to satisfy CCA-security in the standard model. In this section, we discuss difficulty to naturally extend ACE-KEM and FACE-KEM to isogeny-based without changing the structure.

ACE-KEM contains four secret keys $(x_1, x_2, x_3, x_4) \in \mathbb{Z}_p$ and four public keys $(X_1 = g^{x_1}, X_2 = g^{x_2}, X_3 = g^{x_3}, X_4 = g^{x_4})$ where p is a large prime and g is a generator. The ciphertext contains $C_1 = g^r$, $C_2 = X_1^r$ and $C_3 = X_2^r X_3^{r \cdot TCR(C_1, C_2)}$ where $r \in_R \mathbb{Z}_p$ and TCR is a target collision resistance hash function. In decapsulation, validity of the ciphertext is verified by checking if $C_1^{x_2 + x_3 \cdot TCR(C_1, C_2)} = C_3$ holds. It seems that it can be replaced by the CSIDH system as $C_1 = [\mathfrak{r}] * E_0$, $C_2 = [\mathfrak{r}] * X_1$ and $C_3 = ([\mathfrak{r}] * X_2) \cdot ([\mathfrak{r} \cdot TCR(C_1, C_2)] * X_3)$, and checking $[\mathfrak{x}_2 + \mathfrak{x}_3 \cdot TCR(C_1, C_2)] * C_1 = C_3$. However, public keys and the ciphertext are elements in X, and X has no algebraic structure as Definition 4. It means that the relation $([\mathfrak{g}] * E_0) \cdot ([\mathfrak{g}'] * E_0) = [\mathfrak{g} + \mathfrak{g}'] * E_0$ is not guaranteed. Hence, such a decapsulation strategy does not work in the CSIDH system (and also the SIDH system).

FACE-KEM also has a similar situation. FACE-KEM contains four secret keys $(x_1, x_2, y_1, y_2) \in \mathbb{Z}_p$ and two public keys $(X = g_1^{x_1} g_2^{x_2}, Y = g_1^{y_1} g_2^{y_2})$ where g_1 and g_2 are generators. The ciphertext contains $C_1 = g_1^r$, $C_2 = g_2^r$ and $C_3 = T$ such that $(K||T) \leftarrow KDF(X^r Y^{r \cdot TCR(C_1, C_2)})$ where KDF is a key derivation function. In decapsulation, validity of the ciphertext is verified by checking if $T' = C_3$ holds such that $(K'||T') \leftarrow KDF(C_1^{x_1 + y_1 \cdot TCR(C_1, C_2)} \cdot C_2^{x_2 + y_2 \cdot TCR(C_1, C_2)})$. As ACE-KEM, it cannot work in the CSIDH system.

Conversely, PSEC-KEM and ECIES-KEM do not contain such a structure. Also, in security proofs of CSIDH-PSEC-KEM and CSIDH-ECIES-KEM, X does not have to has algebraic structure. Thus, we can extend these KEM schemes to CSIDH-based. Constructing a CCA-secure KEM scheme in the standard model from isogeny is a remaining problem for further research.

References

[ABR99] Michel Abdalla, Mihir Bellare, and Phillip Rogaway. DHAES: An Encryption Scheme Based on the Diffie-Hellman Problem. In *Cryptology ePrint Archive, Report 1999/007*, 1999.

[BGK+18] Dan Boneh, Darren Glass, Daniel Krashen, Kristin Lauter, Shahed Sharif, Alice Silverberg, Mehdi Tibouchi, and Mark Zhandry. Multiparty Non-Interactive Key Exchange and More From Isogenies on Elliptic Curves. In *MATHCRYPT 2018*, 2018. https://eprint.iacr.org/2018/665.

[BODF+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random Oracles in a Quantum World. In *ASIACRYPT 2011*, pages 41–69, 2011.

[CJS14] Andrew M. Childs, David Jao, and Vladimir Soukharev. Constructing elliptic curve isogenies in quantum subexponential time. *J. Mathematical Cryptology*, 8(1):1–29, 2014.

[CLM+18] Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: An Efficient Post-Quantum Commutative Group Action. In *ASIACRYPT*, pages 395–427, 2018.

[Cou06] Jean-Marc Couveignes. Hard Homogeneous Spaces. Cryptology ePrint Archive, Report 2006/291, 2006.

[CS98] Ronald Cramer and Victor Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In *CRYPTO 1998*, pages 13–25, 1998.

[CS04] Ronald Cramer and Victor Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. In *SIAM Journal on Computing 33*, pages 167–226, 2004.

[DG19] Samuel Dobson and Steven D. Galbraith. On the Degree-Insensitive SI-GDH problem and assumption. Cryptology ePrint Archive, Report 2019/929, 2019.

[FO99] Eiichiro Fujisaki and Tatuaki Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. In *CRYPTO 1999*, pages 537–554, 1999.

[FTTY18] Atsushi Fujioka, Katsuyuki Takashima, Shintaro Terada, and Kazuki Yoneyama. Supersingular Isogeny Diffie-Hellman Authenticated Key Exchange. In *ICISC*, pages 177–195, 2018.

[FTY19] Atsushi Fujioka, Katsuyuki Takashima, and Kazuki Yoneyama. One-Round Authenticated Group Key Exchange from Isogenies. In *ProvSec 2019*, 2019. https://eprint.iacr.org/2018/1033.

[GPST16] Steven D. Galbraith, Christophe Petit, Barak Shani, and Yan Bo Ti. On the Security of Supersingular Isogeny Cryptosystems. In *ASIACRYPT (1) 2016*, pages 63–91, 2016.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A Modular Analysis of the Fujisaki-Okamoto Transformation. In *TCC (1) 2017*, pages 341–371, 2017.

[ISO] Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers. ISO/IEC 18033-2:2006 and ISO/IEC 18033-2:2006/AMD1:2017. https://www.iso.org/standard/37971.html.

[SIKE17] David Jao and et al. Supersingular Isogeny Key Encapsulation (SIKE). *submission to NIST PQC Competition*, 2017. https://sike.org/.

[JF11] David Jao and Luca De Feo. Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In *PQCrypto*, pages 19–34, 2011.

[KP14] Kaoru Kurosawa and Le Trieu Phong. Kurosawa-Desmedt Key Encapsulation Mechanism, Revisited. In *AFRICACRYPT 2014*, pages 51–68, 2014.

[NIST] Post-Quantum Cryptography Standardization. National Institute of Standards and Technology. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography.

[NTT08] NTT Corporation. PSEC-KEM Specification version 2.2. *PSEC-KEM website*, 2008. https://info.isl.ntt.co.jp/crypt/eng/psec/.

[Pei19] Chris Peikert. He Gives C-Sieves on the CSIDH. Cryptology ePrint Archive, Report 2019/725, 2019.

[RS06] Alexander Rostovtsev and Anton Stolbunov. Public-Key Cryptosystem Based on Isogenies. Cryptology ePrint Archive, Report 2006/145, 2006.

[Sho94] Peter W. Shor. Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In *ANTS 1994*, page 289, 1994.

[Sho01] Victor Shoup. A Proposal for an ISO Standard for Public Key Encryption. Cryptology ePrint Archive, Report 2001/112, 2001.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004.

[SRP18] Alan Szepieniec, Reza Reyhanitabar, and Bart Preneel. Key Encapsulation from Noisy Key Agreement in the Quantum Random Oracle Model. Cryptology ePrint Archive, Report 2018/884, 2018.

[SXY18] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. Tightly-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model. In *EUROCRYPT (3) 2018*, 2018.

[TU16] Ehsan Ebrahimi Targhi and Dominique Unruh. Post-Quantum Security of the Fujisaki-Okamoto and OAEP Transforms. In *TCC (B2) 2016*, pages 192–216, 2016.

[Vél71] Jacques Vélu. Isogénies entre courbes elliptiques. In *Comptes Rendus de l'Academie des Sciences de Paris*, volume 273, pages A238–A241, 1971.

[XY19] Keita Xagawa and Takashi Yamakawa. (Tightly) QCCA-Secure Key-Encapsulation Mechanism in the Quantum Random Oracle Model. In *PQCrypto 2019*, pages 520–551, 2019.

[Zha12] Mark Zhandry. Secure Identity-Based Encryption in the Quantum Random Oracle Model. In *CRYPTO*, pages 758–775, 2012.