Key Policy Attribute-Based Encryption Leveraging Isogeny-Based Cryptography

Madické Diadji Mbodj[1] and Anis Bkakria[1]

IRT SystemX, Palaiseau 91120, France

Abstract. We present the first Key Policy Attribute-Based Encryption (KP-ABE) scheme employing isogeny-based cryptography through class group actions, specifically utilizing the Csi-FiSh instantiation and pairing groups. We introduce a new assumption, denoted Isog-DLin, which combines the isogeny and DLin assumptions. We propose the following constructions: a small universe KP-ABE and a large universe KP-ABE under the Isog-DBDH assumption, and a small universe KP-ABE under the Isog-DLin assumption. In these constructions, the master key is designed to be secure against quantum computer attacks, while the ciphertext remains secure against classical computer attacks. This dual-layered approach ensures robust security across classical and quantum computational paradigms, addressing current and potential future cryptographic challenges.

Keywords: KP-ABE · Isogeny-Based Cryptography · Isog-DLin

1 Introduction

The confidentiality of private data is receiving increasing attention, highlighting the need to protect this information. One effective method to safeguard data is by implementing a fine-grained access policy. A fine-grained access policy allows access rights to users based on specific conditions, addressing several key areas: security, compliance, operational efficiency, and flexibility.

There are several known techniques for implementing fine-grained access control. Some common techniques include those outlined in [21], [16], and [18]. In this context, attribute-based encryption (ABE), a form of public key encryption (PKE), controls data flows by providing fine-grained access policies. Originally, this scheme was extended from Identity-based Encryption (IBE), first presented by Sahai and Waters [31]. Depending on the policies, there are two types of ABE schemes. The first is key-policy attribute-based encryption (KP-ABE), where a subset of attributes allows the encryption of a message, and the user's private key is linked with an access control policy. This method limits the data owner's ability to determine who can decrypt the data. The second is ciphertext-policy attribute-based encryption (CP-ABE), where the access control policy is embedded into the ciphertext, and the user's private key is associated with attributes.

Many attribute-based encryption (ABE) schemes, such as Bilinear Diffie-Hellman (BDH), rely on pairing-based cryptography. However, with the emergence of quantum computers and algorithms like Shor's [33] and Grover's [17], all

cryptosystems based on hardness assumptions such as discrete logarithms and factorization will become vulnerable to quantum attacks. There is an urgent need to develop post-quantum ABE cryptosystems to address these vulnerabilities. Some post-quantum solutions include lattice-based cryptography, multivariate-based cryptography, isogeny-based cryptography using elliptic curves, hash-based cryptography, and code-based cryptography. Currently, all known post-quantum ABE systems are based on lattice-based cryptography. However, exploring ABE schemes grounded in alternative mathematical foundations is important. For example, a draft NIST report [9] emphasizes the challenge of precisely estimating the security of lattice schemes against known cryptanalysis techniques. Isogenies offer promising attributes, including robustness against quantum attacks, key compactness, computational efficiency, compatibility with existing systems, and application flexibility. These advantages position isogeny-based cryptography as a promising technology for securing communications in the age of quantum computers.

1.1 Related Work

In this subsection, we review related works based on post-quantum ABE. All known post-quantum ABE schemes are based on lattices, with security resting on cryptographic complexity assumptions from Learning with Errors (LWE) to Ring Learning with Errors (RLWE). We begin with KP-ABE and conclude with CP-ABE.

KP-ABE The initial lattice-based KP-ABE scheme was pioneered by Boyen in 2013, as introduced in [5]. This scheme was built upon the hardness of the LWE problem and employed a selective threat model, devoid of collusion, adhering to IND-CPA criteria. Subsequently, Boyen and Li refined the scheme to accommodate finite automata with constrained input sizes. Kuchta and Markowitch [20] utilized their threshold gates to support multiple cloud servers. Zelein [27] created a tree access structure to construct an access policy. Tan and Samsudin extended the LWE problem to the hardness of the decisional RLWE problem. They also developed a KP-ABE scheme with homomorphic encryption to support multi-user cloud environments. Dai et al. [12] used the PALISADE library for a practical KP-ABE implementation. Zhao and Gao [45] improved the LSSS, though the number of secret keys increased exponentially. Yu et al. [42] enhanced the tree structure to support gates like AND, OR, and threshold gates as LSSS. Luo et al. [26] developed proxy encryption to address forward and backward secrecy in this scheme. Pal and Dutta [28] extended the scheme to support functional encryption. In 2023, Luo et al. proposed a revocable attribute-based encryption scheme that supported depth encryption and featured a short secret key, where the key size depended only on the depth of the supported policy function. In 2024, Nejad et al. [22] proposed a post-quantum fuzzy IBE based on the LWE problem, reducing key length and computational complexity during the encryption phase.

CP-ABE One notable aspect of CP-ABE is that it grants users control over encrypted plaintext while also facilitating scalability. In 2012, Zhang et al. [44] extended Sahai and Waters' work [31] to a lattice-based CP-ABE for supporting multi-valued attributes. Zhang and Jiang [43] extended it to q-ary lattices to support multi-bit operations, though this approach faced quadratic overhead issues. Fun and Samsudin [35] resolved the computational overhead of CP-ABE by enhancing RLWE assumptions, though their master secret key remained vulnerable. Zeng and Xu [13] developed the scheme for keyword-searchable functions. To address this issue, Tan and Samsudin [36] added homomorphic encryption with the hardness of RLWE. Fun and Samsudin [14] also studied the scheme using a small universe in the threshold CP-ABE scheme. Yang et al. [41] and Zhao et al. [19] introduced improvements to the CP-ABE scheme by implementing a binary tree structure and threshold gates, respectively. Tsabary [37] devised a CP-ABE scheme based on t-CNF and the LWE problem. Liu et al. [25] addressed user scalability concerns by extending the threshold access structure to support multi-authority levels. Li et al. [1] tackled proxy re-encryption issues in CP-ABE through trapdoor sampling and vector decomposition techniques. Affum et al. [2] explored RLWE-based CP-ABE schemes for supporting 5G content-centric networks. Qian [29] and Wu proposed a basic access tree (BAT) to enhance tree structures, allowing the expression of any disjunctive normal form (DNF). Varri et al. [38] extended CP-ABE to enable searchability over encrypted data.

However, current lattice-based ABE schemes face challenges related to computational complexity and the length of ciphertexts and keys. Yilei Chen et al. [10] show that "we solve the Learning with Errors (LWE) problem with certain polynomial modulus-noise ratios in polynomial time using a quantum algorithm." Many researchers are studying this paper. We are working on post-quantum ABE based on isogenies between supersingular elliptic curves to address these issues. In 2016 and 2019, Koshiba and Takashima developed new frameworks based on isogeny pairing; however, these frameworks are based on SIDH, and their security lies between quantum and classical security [23,24]. In 2016, Galbraith et al. [15] proposed an active attack on the supersingular isogeny encryption scheme of SIDH, showing that the security of these schemes depends on the difficulty of computing the endomorphism ring of a supersingular elliptic curve. They provided a reduction that uses partial knowledge of shared keys to determine the entire shared key. On August 5, 2022, Castryck and Decru posted a preprint [7] demonstrating the vulnerability of SIDH, rendering all its variants insecure. Consequently, schemes proposed based on isogeny pairing groups by Koshiba and Takashima are not secure. [1].

1.2 Contribution

This paper introduces the first Attribute-Based Encryption (ABE) scheme based on isogeny group actions of the ideal class group, utilizing Csi-FiSh. Following the structure of [23], we employ the isogeny pairing group framework and its

[1] For more information on post-quantum ABE based on lattices: [32]

associated intractability assumptions. We propose a Key-Policy Attribute-Based Encryption (KP-ABE) scheme where key generation is secure against quantum adversaries, and ciphertext is secure against classical adversaries. The security of our scheme is achieved as follows:

- To address the limitations of the schemes in [23,24], which utilize a trapdoor homomorphism where the isogeny is based on ideal class group action (CSIDH), we leverage Csi-FiSh. This approach enables us to efficiently evaluate the ideal class group and apply the discrete logarithm with the ideal generator.
- We propose both small and large universe KP-ABE schemes. The modification of the public and master keys involves the isogeny of supersingular elliptic curves, based on Csi-FiSh.
- Initially, we adopt the Isog-DBDH security assumption (a combination of isogeny and decisional bilinear Diffie-Hellman assumptions) and introduce a new assumption, denoted Isog-DLin (a combination of isogeny and decisional linear assumptions), under which we construct a small universe KP-ABE scheme.
- Our schemes are secure in a selective security model. Security is demonstrated through a game-based proof between an adversary $\mathcal{A}$ and their challenger $\mathcal{B}$, conducted in two phases:
  - → First, the adversary $\mathcal{A}$ analyzes the public key and attempts to extract information from the master secret key, which is protected by the hard assumption of isogeny of class group action.
  - → Second, the adversary tries to distinguish between the two encrypted messages provided by $\mathcal{B}$.

  In summary, the security game differentiates between quantum security and classical security. In the context of classical security, the scheme is resistant to collusion.

Remark 1. We have explored the full definition of post-quantum KP-ABE. However, we conclude that given the specific nature of attribute-based encryption, isogenies do not currently provide the necessary components to construct a complete post-quantum ABE. The building blocks required are still lacking.

1.3 Paper Organisation

This paper is organized as follows: In section 2, we provide essential pieces of information about access structures, trapdoor homomorphisms, the mathematical background on isogenies, and security definitions of isogeny pairing groups. We present new security definitions, denoted Isog-DLin, in section 3. In section 4, we present our constructions of a small universe KP-ABE under the Isog-DBDH and Isog-DLin assumptions, and in section 5, we detail a large universe KP-ABE. section 6 provides the security analysis of the small and large universe KP-ABE constructions. Finally, section 6 concludes the paper.

2 Preliminaries

This section introduces some basic definitions and mathematical background that we will use for our construction. Before that, let's present some notations that we use throughout this paper.

2.1 Notations

- Let E be a supersingular elliptic curve over $\mathbb{F}_p$. The group of set points of E over $\mathbb{F}_p$ is denoted $\mathsf{E}(\mathbb{F}_p)$. A cyclic subgroup of $\mathsf{E}(\mathbb{F}_p)$ is denoted $\mathbb{G}$ and g is an element of $\mathbb{G}$.
- When A is a random variable or a distribution, $y \xleftarrow{\$} A$ means y is randomly selected from A according to its distribution.
- Let $[n] := \{1, \ldots, n\}$ and $[0, n] := \{0, \ldots, n\}$ for any positive integer n.
- For two vectors $\vec{y} = (y_i)_{i\in[r]}$ and $\vec{v} = (v_i)_{i\in[r]}$, $\vec{y}.\vec{v}$ denotes the inner product $\sum_{i=1}^{r} y_i v_i$.

2.2 Definitions

This subsection presents some basic definitions for access structures and secret sharing schemes.

Definition 1 (Access structure). *Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbf{A} \subseteq 2^{\{P_1,\ldots,P_n\}}$ is monotone if $\forall B, C: \quad B \in \mathbf{A}, B \subseteq C$, then $C \in \mathbf{A}$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) $\mathbf{A}$ of non-empty subsets of $\{P_1, \ldots, P_n\}$, ie, $\mathbf{A} \subseteq 2^{\{P_1,\ldots,P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbf{A}$ are called the authorized sets, and the sets not in $\mathbf{A}$ are called the unauthorized sets. In this context, monotone means that an authorized user who acquires more attributes will not lose privileges.*

Definition 2 (span program [3]). *A span program over $\mathbb{F}_p$ is a labeled matrix $\mathbb{S} := (M, \rho)$ where M is a matrix of $l \times r$ over $\mathbb{F}_p$ and ρ is a labeling of the rows of M by an attribute from $\{(t, v), (t', v'), \cdots\}$ (every row is labeled by one attribute), i.e., from $\{1, \cdots, l\} \to \{(t, v), (t', v'), \cdots\}$. Let $\Gamma := \{(t_j, x_j)\}_{1\le j\le d'}(x_j \in \mathcal{U}_{t_j})$. The span program $\mathbb{S}$ accepts Γ if only if $\vec{1} \in$ span $\langle (M_i)_{\rho(i)\in\Gamma}\rangle$, i.e., some linear combination of the rows $(M_i)_{\rho(i)\in\Gamma}$ gives all one vector $\vec{1}$.*

Definition 3 (Secret Sharing Scheme for Span Program $\mathbb{S} := (M, \rho)$). *Let M be an $l \times r$ matrix and ρ a labeling of the rows of M. A secret sharing scheme for the span program $\mathbb{S} := (M, \rho)$ consists of:*

1. *A random vector $\vec{u} \xleftarrow{\$} \mathbb{F}_p^r$ such that $\vec{1} \cdot \vec{u} = s$, where s is the secret to be shared, and $(u_i) \xleftarrow{\$} \mathbb{F}_p^r$. Then, $\vec{s} := (s_1, \ldots, s_l) = M \cdot \vec{u}^\top$ represents the l shares of the secret s, and each s_i belongs to $\rho(i)$.*
2. *If the span program $\mathbb{S}$ accepts Γ (i.e., $\vec{1} \in$ span$\langle M_i \mid \rho(i) \in \Gamma\rangle$), there exist constants $\{\sigma_i \in \mathbb{F}_q \mid i \in I\}$ such that $I \subseteq \{i \in [l] \mid \rho(i) \in \Gamma\}$ and $\sum_{i\in I} \sigma_i s_i = s$. Furthermore, these constants $\{\sigma_i\}$ can be computed in time polynomial in the size of the matrix M.*

2.3 KP-ABE

A **KP-ABE** consists of four algorithms (Setup, KeyGen, Encrypt, Decrypt).

- Setup is a quantum ppt algorithm. It takes an input security parameter λ and outputs a public key pk and a master secret key msk.
- KeyGen is a ppt algorithm. It takes an input pk, msk, and access structure $\mathbb{S} = (M, \rho)$ and outputs a secret key sk.
- Encrypt is ppt algorithm. It takes an input pk, a plaintext m, and a subset of attributes Γ and outputs a ciphertext c_T.
- Decrypt is a deterministic algorithm. It takes as input pk, c_T, sk, where the ciphertext c_T is associated with the set of attributes Γ of the user encryptor. If Γ is accepted by the access structure $\mathbb{S}$, then the algorithm outputs the plaintext; otherwise, it outputs an error symbol $\perp$.

2.4 Mathematical background on isogenies

Elliptic curves possess mathematical properties that allow efficient computational time and small key size. Generally, for a field $\mathbb{K}$ with algebraic closure $\bar{\mathbb{K}}$, an elliptic curve contains points $(x, y) \in \bar{\mathbb{K}}^2$ that satisfy affine curve equation

$$\mathsf{E}: \; y^2 + a_1xy + a_3y \; = \; x^3 + a_2x^2 + a_4x + a_6$$

where $(a_1, \cdots, a_6) \; \in \bar{\mathbb{K}}$ and $\mathcal{O} = \; [0, 1, 0]$ denoted point at infinity [2]. For a large prime p, if an elliptic curve E defined over $\mathbb{F}_p$ is supersingular then, the cardinal $\#\mathsf{E}(\mathbb{F}_p) = p + 1$. Following the properties as mentioned earlier, an isogeny between two elliptic curves $\mathsf{E}_1, \mathsf{E}_2$ is a non-constant morphism $\phi \; : \mathsf{E}_1 \longrightarrow \mathsf{E}_2$ that satisfies $\phi(\infty) = \infty$. The equation for E_2 and the isogeny ϕ can be computed using the Vélu formula [39]. $\hat{\phi} : \mathsf{E}_2 \to \mathsf{E}_1$ is the dual ϕ. In this way, there exist two popular key exchange cryptographic based on isogeny: Supersingular Isogeny Diffie-Hellman (SIDH)[20] and Commutative Supersingular Isogeny Diffie-Hellman (CSIDH) [8]. SIDH was broken by Castryck et al [7].

CSIDH is a key exchange protocol that Castryck et al. introduced [8] using isogenies of ideal class group actions. It builds on the work of Couveignes [11], who introduced the notion of Hard Homogeneous Spaces (HHS), which have properties like vectorization and parallelization that protect quantum algorithms, following the ideas of Rostovtsev and Stolbunov [30]. Couveignes Rostovtsev and Stolbunov (C.R.S) worked on ordinary elliptic curves. Efficient algorithms are needed to construct an HHS to evaluate a class group, where isogeny plays a crucial role.

CSIDH is based on actions of the ideal class group denoted $\mathcal{C}l(\mathbb{Z}[\pi_p])$(see [3]) on $\mathbb{Z}[\pi_p]$ (on $\mathcal{E}ll_p(\mathcal{O})$ see [4]), in which $p = 4.l_1 \cdots l_n - 1$ is a large prime. $l_1 \cdots l_n$ are

[2] For more about elliptic curves see Silverman's book [34]

[3] $\mathcal{C}l(\mathbb{Z}[\pi_p])$ is the ideal class group of the endomorphims ring in which it acts on $\mathbb{F}_p$.

[4] $\mathcal{O}$ is an order of quadratic field, $\mathcal{E}ll_p(\mathcal{O})$ denote the set of elliptic curves E defined over $\mathbb{F}_p$ with $\mathrm{End}_p(\mathsf{E}) \cong \mathcal{O}$ such that π corresponds to the $\mathbb{F}_p$-Frobenius endomorphism of E, $[\mathfrak{l}_i^{-1}] = [\bar{\mathfrak{l}}_i]$

small odd primes. Integral ideals $\mathfrak{l}_i (i = 1 \cdots n) \in \mathbb{Z}[\pi_p]$ be $(l_i, \pi_p - 1)$ and Integral ideals $\bar{\mathfrak{l}}_i (i = 1 \cdots n) \in \mathbb{Z}[\pi_p]$ be $(l_i, \pi_p + 1)$ (parametrisation CSIDH 512 see in [6] section 8.1). Let $\mathfrak{a}$ a randomly sample from $\mathcal{C}\mathrm{l}(\mathcal{O})$ in which, $\mathfrak{a} = \mathfrak{l}_1^{e_1}.\mathfrak{l}_2^{e_2} \cdots \mathfrak{l}_n^{e_n}$, where $\{e_1 \cdots e_n\}$ are small integers from range $\{-m, \cdots, m\}$ and m satisfies $2m + 1 \leq \sqrt[n]{\#\mathcal{C}\mathrm{l}(\mathcal{O})}$. So, we have:

$$\begin{aligned} \mathcal{C}\mathrm{l}(\mathbb{Z}_p) \times \mathcal{E}ll_p(\mathcal{O}) &\longrightarrow \mathcal{E}ll_p(\mathcal{O}) \\ (\mathfrak{a}, \mathsf{E}) &\longmapsto \mathsf{E}/\mathsf{E}[\mathfrak{a}] = [\mathfrak{a}]\mathsf{E}. \end{aligned}$$

Alice		**Bob**
$([\mathfrak{a}], A)$		$([\mathfrak{b}], B)\ B \in \mathbb{F}_p\ \{\pm 2\}$
$\mathsf{E}_A : y^2 = x^3 + Ax^2 + x$		$\mathsf{E}_B : y^2 = x^3 + Bx^2 + x$
	$\{[\mathfrak{a}]\mathsf{E}_0 = \mathsf{E}_A\} \longrightarrow$	
	$\{[\mathfrak{b}]\mathsf{E}_0 = \mathsf{E}_B\} \longleftarrow$	
$\mathsf{E}_B \in \mathcal{E}ll_p(\mathcal{O})?$		$\mathsf{E}_A \in \mathcal{E}ll_p(\mathcal{O})?$
$[\mathfrak{a}]\mathsf{E}_B = [\mathfrak{a}][\mathfrak{b}]\mathsf{E}_0$		$[\mathfrak{b}]\mathsf{E}_A = [\mathfrak{b}][\mathfrak{a}]\mathsf{E}_0$
	$[\mathfrak{a}][\mathfrak{b}]\mathsf{E}_0 = [\mathfrak{b}][\mathfrak{a}]\mathsf{E}_0$	
	$\mathsf{E}_S :\ y^2 = x^3 + \mathbf{S}x^2 + x$	

The shared secret is the Montgomery coefficient $\mathbf{S}$ of the common secret curve $\mathsf{E}_\mathbf{S}$.

Normally, CSIDH is designed to evaluate $\mathcal{C}\mathrm{l}(\mathcal{O})$ as efficiently as possible. However, even though all axioms of HHS are satisfied, it is not possible to efficiently evaluate the action of any element of $\mathcal{C}\mathrm{l}(\mathbb{Z}_p)$, nor is it possible to verify the equality of two elements of $\mathcal{C}\mathrm{l}(\mathbb{Z}_p)$. To overcome the previous limitation, Beullens et al. [4] published a signature and identification scheme based on class group computation called Csi-Fish using CSIDH 512[4]. In Csi-Fish [4], all ideals are assumed to generate the class group $\mathcal{C}\mathrm{l}(\mathcal{O})$, and in practice, we choose one $\mathfrak{l}_i$ to generate the class group. In fact, for the CSIDH 512 class group, we can even take $\mathfrak{l}_1 = \langle 3, \pi - 1 \rangle$. Thus, there exists a lattice relation $\mathcal{L} := e = (e_1, \cdots, e_n) \in \mathbb{Z}^n : \prod_{i=1}^n \mathfrak{l}_i^{e_i} = (1)$, which yields a representation of a class group as $\mathcal{C}\mathrm{l}(\mathbb{Z}[\pi]) \simeq \mathbb{Z}^n/\mathcal{L}$. The equality of two vectors can be tested by checking if $e - f \in \mathcal{L}$. By solving the approximate closest vector problem (CVP) for $f \in \mathcal{L}$ and evaluating $e - f$, we can obtain the ideal $\mathfrak{a} = \prod_{i=1}^n \mathfrak{l}_i^{e_i}$. This facilitates evaluating $\mathcal{C}\mathrm{l}(\mathcal{O})$ and verifying elements within $\mathcal{C}\mathrm{l}(\mathcal{O})$.

2.5 Trapdoor homomorphism

This section presents some definitions of trapdoor homomorphism and its properties, starting with the Group Action Inverse Problem definition.

Definition 4 (Group Action Inverse Problem (GAIP) [4]). *Given two supersingular curves* E, E' *with* $End(\mathsf{E}) = End(\mathsf{E}') = \mathcal{O}$ *and* $\mathfrak{g}$ *a generator of* $\mathcal{C}l(\mathcal{O})$*. The GAIP is to find* $\mathfrak{a}$ *such that* $\mathsf{E}' := \mathfrak{g}^a \star \mathsf{E}$, $([\mathfrak{a}] = \mathfrak{g}^a)$.

Definition 5 (Trapdoor Homomorphisms (TH) [23]). *A (randomly chosen) function* $\phi := \phi_\xi : \mathbb{G}_0 \to \mathbb{G}_1$ *with two (randomly chosen) cyclic groups* $\mathbb{G}_0, \mathbb{G}_1$ *of a prime order* p *is called a trapdoor homomorphism if the following conditions hold:*

- ϕ *is non-trivial (e.g., non-zero for an additive group) homomorphism.*
- *Intractability assumption: any probabilistic polynomial-time (ppt) machine* $\mathcal{B}$ *computes* $\phi(g)$ *only with a negligible probability when given* $(g_0, \phi(g_0), g)$ *for a randomly chosen* ϕ *and* $g_0, g \xleftarrow{\$} \mathbb{G}_0$.
- *Polynomial-size trapdoor: there exists a probabilistic polynomial time (ppt) machine* B *which computes* $\phi(g)$ *for any* $g \in \mathbb{G}_0$ *given a polynomial-size trapdoor* ξ *for* $\phi := \phi_\xi$.

2.6 Isogeny Pairing Group (IPG)[23]

In this paper, we utilize the Weil pairing [34], denoted as $\mathsf{E}[N] \times \mathsf{E}[N] \to \mu_N$, to describe our protocol. These pairings are embedded in a degree-3 context. An isogenous pairing group consists of a random instance as follows:

$$\mathsf{Gen}^{\mathsf{IPG}} \xrightarrow{\$} \mathrm{pk}^{\mathrm{IPG}} := \left((\mathbb{G}_t, \hat{\mathbb{G}}_t, g_t, \hat{g}_t, e_t)_{t\in[0,d]}, \mathrm{sk}^{\mathrm{IPG}} := (\phi_t)_{t\in[d]} \right),$$

where $(\mathbb{G}_t, \hat{\mathbb{G}}_t, e_t, \mathbb{G}_T)$ is an asymmetric pairing group of prime order p, with pairings $e_t : \mathbb{G}_t \times \hat{\mathbb{G}}_t \to \mathbb{G}_T$, and trapdoor homomorphisms $\phi_t : \mathbb{G}_0 \to \mathbb{G}_t$ (given by isogenies between different elliptic curves), and $\phi_t(g_0) = g_t \in \mathbb{G}_t$. The isogenous pairing groups satisfy the following compatibility property:

$$e_0(g_0, \hat{g}_0) = e_t(g_t, \hat{g}_t) = e_t(\phi_t(g_0), \hat{g}_t), \quad \forall t \in [d].$$

Definition 6 (d-pIsog-DBDH Assumption on IPG [23]).

Let $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ *be the adversary and* $\mathcal{Ch}$ *be their challenger, where* $\mathcal{B}_1$ *is modeled as a polynomial-time quantum adversary, and* $\mathcal{B}_2$ *is a classical probabilistic polynomial-time algorithm. Let* λ *be the security parameter, and* d *be the size of the small universe.*

1. $\mathcal{Ch}$ *computes* $\mathrm{pk}^{\mathrm{IPG}} := ((\mathbb{G}_t, \hat{\mathbb{G}}_t, g_t, \hat{g}_t, e_t)_{t\in[0,d]}, h_T, \mathbb{G}_T), \mathsf{sk}^{IPG} := (\phi_t)_{t\in d} \xleftarrow{\$} \mathsf{Gen}^{\mathsf{IPG}}(\lambda, d)$
2. $\mathcal{Ch}$ *sends* $\mathrm{pk}^{\mathrm{IPG}}$ *to* $\mathcal{B}_1$*;*
3. $\mathcal{B}_1$*, with input* $\mathrm{pk}^{\mathrm{IPG}}$*, outputs some* state*;*

4. $\mathtt{Ch}$ *samples* $(\alpha, \beta, \delta) \xleftarrow{\$} \mathbb{F}_p$ *and computes*

$$\mathcal{X}_0 := (\text{state}, g_0^{\alpha}, (\hat{g}_t^{\beta})_{t\in[d]}, g_T^{\alpha\beta}),$$
and
$$\mathcal{X}_1 := (\text{state}, g_0^{\alpha}, (\hat{g}_t^{\beta})_{t\in[d]}, g_T^{\gamma}),$$

5. $\mathtt{Ch}$ *sends* $\mathcal{X}_b$ *to* $\mathcal{B}_2$*, where* $b \in \{0,1\}$*;*
6. $\mathcal{B}_2$ *flips a random coin to generate a bit* b'*, and returns* b'*. If* $b = b'$*, then* $\mathcal{B}$ *wins;*

We define the advantage of $\mathcal{B}$ *in this security definition as:*

$$\text{Adv}_{\mathcal{B}}^{\textit{d-pIsog-DBDH}}(\lambda) := \Pr[\mathcal{B}\ \textit{wins}] - \frac{1}{2}.$$

The d-pIsog-DBDH assumption is secure against a probabilistic polynomial-time adversary $\mathcal{B}$ *in this experiment if the advantage of* $\mathcal{B}$*,* $\text{Adv}_{\mathcal{B}}^{\textit{d-pIsog-DBDH}}(\lambda)$*, is negligible in* λ*.*

Note that the description of the algorithm $\mathsf{Gen}^{\mathsf{IPG}}$ is in appendix A.1.

Definition 7 (Payload Hiding Pre-Challenge Quantum (PH-PQ) for KP-ABE [23]).
A PH-PQ consists of four algorithms $(\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Decrypt})$ *of a KP-ABE scheme and an adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$*, where* $\mathcal{A}_1$ *is modeled as a polynomial-time quantum adversary with their challenger* $\mathtt{Ch}$ *. We consider the experiment* $\mathsf{Exp}_{\mathcal{A}}^{\textit{KP-ABE, PH-PQ}}[\lambda]$ *as follows:*

1. *The adversary provides* $\Gamma^* \xleftarrow{\$} \mathcal{A}_1(\lambda)$ *to* $\mathtt{Ch}$ *;*
2. $\mathtt{Ch}$ *computes* $(\mathsf{sk}, \mathsf{pk}) \xleftarrow{\$} \mathsf{Setup}(\lambda)$ *and sends* pk *to* $\mathcal{A}_1$*;*
3. $\mathcal{A}_1$ *receives* pk *and outputs* $\text{state} \xleftarrow{\$} \mathcal{A}_1^{RO(\cdot),\mathsf{Gen}(\mathsf{sk},\cdot)}(\mathsf{pk})$ *to* $\mathcal{A}_2$*;*
4. $\mathcal{A}_2$ *chooses* $(m_0, m_1) \xleftarrow{\$} \mathcal{A}_2^{RO(\cdot),\mathsf{Gen}(\mathsf{sk},\cdot)}(\text{state})$*, where* m_0 *and* m_1 *are of the same size;*
5. $\mathtt{Ch}$ *chooses* $b \xleftarrow{\$} \{0,1\}$*, computes* $ct^* \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, m_b, \Gamma^*)$*, and sends* ct^* *to* $\mathcal{A}_2$*;*
6. $\mathcal{A}_2$ *flips a random coin to generate a bit* b'*.*

RO is a random quantum oracle, and Gen *is classical-accessible. If* $b = b'$*, then* $\mathcal{A}$ *wins.*
We define the advantage of $\mathcal{A}$ *in this security definition as:*

$$\text{Adv}_{\mathcal{A}}^{\textit{KP-ABE, PH-PQ}}(\lambda) := \Pr[\mathcal{A}\ \textit{wins}] - \frac{1}{2}.$$

A KP-ABE scheme achieves payload hiding against a pre-challenge quantum adversary if, for all adversaries $\mathcal{A}$*, the advantage* $\text{Adv}_{\mathcal{A}}^{\textit{KP-ABE, PH-PQ}}(\lambda)$ *is negligible in* λ*.*

Definition 8 (Selective model for KP-ABE).

- Init*: The adversary declares the set of attributes, Γ^*, that they wish to be challenged upon.*
- Setup*: The challenger runs the KP-ABE* Setup *algorithm and provides the adversary with the public parameters.*
- *Phase 1: The adversary is allowed to issue queries for private keys for various access structures $\mathbb{S}_j$, ensuring $\Gamma^* \notin \mathbb{S}_j$ for all j.*
  Challenge: The adversary submits two messages of equal length, m_0 and m_1. The challenger flips a random coin b and encrypts m_b with Γ^*. The resulting ciphertext is given to the adversary.
- *Phase 2: Phase 1 is repeated.*
- *Guess: The adversary submits a guess b' for b.*
  The advantage of an adversary $\mathcal{A}$ in this game is defined as $\Pr[b' = b] - \frac{1}{2}$.

3 Security definitions

This section introduces the definitions of security that we use to construct a KP-ABE.

Based on the framework of Waters [40], as defined by the DLin assumption, we propose a new hypothesis that blends isogeny and the DLin assumption called Isog-DLin. The Isog-DLin assumption is as follows.

Definition 9 (Isog-DLin). *Given a cyclic symmetric pairing group $e : \mathbb{G}_t \times \mathbb{G}_t \to \mathbb{G}_T$, a random isogeny $\phi_t : \mathbb{G}_0 \to \mathbb{G}_t$, $g_0, g, h \xleftarrow{\$} \mathbb{G}_0$, and $(\alpha, \beta) \xleftarrow{\$} \mathbb{F}_p$, the ppt adversary can only guess whether $h_t = \phi_t(g)^{\alpha+\beta}$ or a random element in $\mathbb{G}_t$ with negligible probability, given $(g_0, \phi_t(g_0)^{\alpha}, g, h, \phi_t(h)^{\beta})$.*

Depending on the size of the small universe represented by the value d, we provide three definitions of Isog-DLin on the isogeny pairing group (IPG) in the following lines.

Definition 10 (Isog-DLin Assumption on IPG). *For $d = 1$, let $\mathcal{B}$ be a classical probabilistic polynomial-time (PPT) and $\mathcal{Ch}$ his challenger. Given the public key $\mathsf{pk}^{IPG} := ((\mathbb{G}_t, \hat{\mathbb{G}}_t, g_t, \hat{g}_t, e_t)_{t\in[0,1]}, h_T, \mathbb{G}_T)$, the secret key $\mathsf{sk}^{IPG} := (\phi_1) \xleftarrow{\$} \mathsf{Gen}^{\mathsf{IPG}}(\lambda, 1)$, and random elements $\alpha, \beta, \gamma, \delta \xleftarrow{\$} \mathbb{F}_q$, $\mathcal{B}$ is given the challenge $\mathcal{X}_b$ for $b \xleftarrow{\$} \{0, 1\}$, defined as:*

$$\mathcal{X}_0 := (\mathsf{pk}^{IPG}, g_0^{\alpha}, g_0^{\beta}, \hat{g}_1^{\alpha}, \hat{g}_1^{\beta}, \hat{g}_1^{\gamma}, g_T^{\alpha\gamma+\beta\gamma})$$

and

$$\mathcal{X}_1 := (\mathsf{pk}^{IPG}, g_0^{\alpha}, g_0^{\beta}, \hat{g}_1^{\alpha}, \hat{g}_1^{\beta}, \hat{g}_1^{\gamma}, g_T^{\delta}),$$

where $g_T := e_0(g_0, \hat{g}_0)$. The adversary $\mathcal{B}$ outputs a bit b'. If $b = b'$, then $\mathcal{B}$ wins. The advantage of any PPT adversary $\mathcal{B}$ against the Isog-DLin assumption is negligible in λ.

Noting that the challenge $(\mathcal{X}_b)_{b\in\{0,1\}}$ contains the elements $(g_0, g_0^\alpha, g_0^\beta, \hat{g}_1, \hat{g}_1^\alpha, \hat{g}_1^\beta, \hat{g}_1^\gamma)$, the key question is whether $g_0^{\alpha+\beta} \in \mathbb{G}_0$ is a specific element or a random element in $\mathbb{G}_0$. This question extends to whether $g_T^{\alpha\gamma+\beta\gamma} \in \mathbb{G}_T$ is a specific or random element in $\mathbb{G}_T$. The adversary $\mathcal{B}$ cannot derive the pairing value $g_T^{\alpha\gamma+\beta\gamma}$ from the given elements $g_0^\alpha, g_0^\beta, \hat{g}_1^\alpha, \hat{g}_1^\beta, \hat{g}_1^\gamma$, and $\hat{g}_1^\gamma = \phi_1(\hat{g}_0)^\gamma$.

In the following definitions, we introduce an adversary $\mathcal{B}$ modeled as $\mathcal{B}_1$, a quantum adversary, and $\mathcal{B}_2$, a classical adversary. These definitions differ based on the size of the small universe d.

Definition 11 (pIsog-DLin assumption on IPG). *For $d = 1$, let $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ be an adversary, where $\mathcal{B}_1$ is modeled as a polynomial-time quantum adversary, and $\mathcal{B}_2$ as a classical adversary.*

1. *$\mathtt{Ch}$ computes* $\mathrm{pk}^{\mathrm{IPG}} := ((\mathbb{G}_t, \hat{\mathbb{G}}_t, g_t, \hat{g}_t, e_t)_{t\in[0,1]}, h_T, \mathbb{G}_T)$, $\mathsf{sk}^{IPG} := (\phi_1) \xleftarrow{\$} \mathsf{Gen}^{\mathsf{IPG}}(\lambda, 1)$, *and sends* pk^{IPG} *to* $\mathcal{B}_1$*;*
2. *$\mathcal{B}_1$ outputs state* $\xleftarrow{\$} \mathcal{B}_1(\mathrm{pk}^{\mathrm{IPG}})$*;*
3. *$\mathtt{Ch}$ samples* $\alpha, \beta, \gamma, \delta \xleftarrow{\$} \mathbb{F}_p$ *and provides* $\mathcal{X}_b$ *for* $b \xleftarrow{\$} \{0,1\}$*;*
4. *$\mathcal{B}_2$ receives* $\mathcal{X}_b$ *for* $b \xleftarrow{\$} \{0,1\}$*, as defined by definition 10.* $\mathcal{B}_2$ *outputs a bit* b'*.*

If $b = b'$, then $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ wins. For any adversary, $\mathcal{B}$, the advantage of $\mathcal{B}$ against the Isog-DLin problem is negligible in the security parameter λ.

Definition 12 (d-pIsog-DLin assumption on IPG). *Let $d > 1$ be the size of the universe, and let $\mathcal{B} = (\mathcal{B}_1, \mathcal{B}_2)$ be an adversary, where $\mathcal{B}_1$ is modeled as a polynomial-time quantum adversary and $\mathcal{B}_2$ as a classical adversary. $\mathtt{Ch}$ is their challenger.*

1. *$\mathtt{Ch}$ computes* $(\mathrm{pk}^{\mathrm{IPG}} := ((\mathbb{G}_t, \hat{\mathbb{G}}_t, g_t, \hat{g}_t, e_t)_{t\in[0,d]}, \mathbb{G}_T)$, $\mathrm{sk}^{\mathrm{IPG}} := (\phi_t)_{t\in d}) \xleftarrow{\$} \mathsf{Gen}^{\mathsf{IPG}}(\lambda, d)$*;*
2. *$\mathtt{Ch}$ sends* $\mathrm{pk}^{\mathrm{IPG}}$ *to* $\mathcal{B}_1$*;*
3. *$\mathcal{B}_1$ outputs state* $\xleftarrow{\$} \mathcal{B}_1(\mathsf{pk}^{IPG})$*;*
4. *$\mathtt{Ch}$ samples* $(\alpha, \beta, \delta) \xleftarrow{\$} \mathbb{F}_p$*;*
5. *$\mathtt{Ch}$ computes*

$$\mathcal{X}_0 := (\mathsf{state}, g_0^\alpha, g_0^\beta, (\hat{g}_t^\alpha, \hat{g}_t^\beta, \hat{g}_t^\gamma)_{t\in[d]}, g_T^{\alpha\gamma+\beta\gamma}),$$

and

$$\mathcal{X}_1 := (\mathsf{state}, g_0^\alpha, g_0^\beta, (\hat{g}_t^\alpha, \hat{g}_t^\beta, \hat{g}_t^\gamma)_{t\in[d]}, g_T^\delta),$$

where $g_T := e_0(g_0, \hat{g}_0)$. $\mathcal{B}$ outputs a bit b'. If $b = b'$, $\mathcal{B}$ wins.
The advantage of the adversary $\mathcal{B}$ in the experiment is defined as:

$$\mathrm{Adv}_{\mathcal{B}}^{d\text{-}pIsog\text{-}DLin}(\lambda) := \Pr[\mathcal{B}\ wins] - \frac{1}{2}.$$

The d-pIsog-DLin assumption is secure against the probabilistic polynomial-time adversary $\mathcal{B}$ if the advantage of $\mathcal{B}$, $\mathrm{Adv}_{\mathcal{B}}^{d\text{-}pIsog\text{-}DLin}(\lambda)$, is negligible in λ.

4 Small universe KP-ABE

For a polynomial $d = d(\lambda)$, a sub-universe $\mathcal{U}_t (\subset \{0,1\}^*)$ is assigned for $t \in [d]$. Each attribute is expressed by a pair (t, v), where $t \in [d]$ and $v \in \mathcal{U}_t$. Let $\mathcal{U}_t := \{1\}$ for a small universe case and $\mathcal{U}_t := \{0,1\}^n$. The **IPG** gives $(d+1)$ pairing groups.

4.1 Construction under Isog-DBDH assumption

The public key pk is vulnerable to quantum attacks in this construction. To protect it, we encode group elements in different groups (i.e., different elliptic curves). For example, for $(g_t \in \mathbb{G}_t)_{t\in[d]}$, $g_t := \phi_t(g_0) \in \mathbb{G}_t := \phi_t(\mathbb{G}_0)$. Note that we should not include two or more different elements in the same group because if we include two or more elements in the same group, the quantum adversary could find the exponent. In this paper, $t \in [d]$ and T is different from t, and all pairings map to $\mathbb{G}_T$.

$\underline{\mathsf{Setup}(\lambda, d)}$:
Input: (λ, d)
Output: $(\mathsf{pk}, \mathsf{msk})$

1: Choose $\mathsf{E}_0 : y^2 = x^3 + x^2$ ▷ a supersingular curves
2: Choose $g_0 \xleftarrow{\$} \mathbb{G}_0$, $\hat{g}_0 \xleftarrow{\$} \hat{G}_0$ ▷ $\mathbb{G}_0$ and $\hat{G}_0 \in \mathsf{E}_0(\mathbb{F}_p)$
3: $(\mathsf{E}_t, \zeta) \xleftarrow{\$} \mathrm{Isog}_{3,k}(\mathsf{E}_0)$.
4: $h_T \xleftarrow{\$} \mathbb{G}_T$ such that $h_T := e_0(g_0, \hat{g}_0)$
5: $\mathsf{pk} := ((\mathbb{G}_t, \hat{G}_t, \hat{g}_t, e_t)_{t\in[d]}, \mathbb{G}_T, h_T)$; $\mathsf{msk} := (\phi_t)_{t\in[d]}$)
6: **return** $(\mathsf{pk}, \mathsf{msk})$

$\underline{\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, \mathbb{S} = (M, \rho))}$:
Input: $(\mathsf{pk}, \mathsf{msk}, \mathbb{S} = (M, \rho))$
Output: $\mathsf{sk}_{\mathbb{S}}$

1: Choose a random vector $\vec{u} \xleftarrow{\$} \mathbb{F}_q^r$ such that $\vec{1}.\vec{u} = \sum_{i=1}^{r} u_i = s$ where s is secret to be shared and $(u_i) \xleftarrow{\$} \mathbb{F}_p^r$. There exists h'_0 where $e_0(h'_0, \hat{g}_0) = h_T^{s^{-1}}$
2: $M_i.\vec{u}^\top$, $t \in \rho(i)$
3: $k_i := \phi_t(h'_0)^{s_i}$
4: **return** $sk_{\mathbb{S}} := \{k_i\}_{i\in[l]}$

$\underline{\mathsf{Encrypt}(\mathsf{pk}, m, \Gamma)}$:
Input: (pk, m, Γ)
Output: $c_T := (\{c_t\}_{t\in\Gamma}, c, \Gamma)$

1: $\zeta \xleftarrow{\$} \mathbb{F}_p$
2: $\forall t \in \Gamma$, $c_t := \hat{g}_t^{\zeta}$
3: $z := h_T^{\zeta} = e_0(g_0, \hat{g}_0)^{\zeta}$
4: $c = m.z$
5: **return** $c_T := (\{c_t\}_{t\in\Gamma}, c, \Gamma)$

$\underline{\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}_{\mathbb{S}}, c_T)}$:
Input: $((\mathsf{pk}, \mathsf{sk}_{\mathbb{S}}, c_T)$
Output: m' or $\perp$

1: **if** Γ satisfies $\mathbb{S}$ **then**
2: **for** $\rho(i) \in \Gamma$ **do**
3: computes σ_i such that $\vec{1} = \sum_{\rho(i)\in\Gamma} \sigma_i.M_i$
4: $z' := \prod_{t:=\rho(i)\in\Gamma} e_t(k_i, c_t)^{\sigma_i}$
return $m' := c.z'^{-1}$
5: **else**
6: **return** $\perp$

Note that the description of the algorithm $\mathrm{Isog}_{3,k}$ is in appendix A.

4.2 Construction under Isog-DLin assumptions

As seen in the previous section, the Setup and KeyGen procedures are identical. The construction is as follows:

Encrypt(pk, m, Γ):
Input: (pk, m, Γ)
Output: $c_T := (\{c_t\}_{t\in\Gamma}, c, \Gamma)$

1: $\zeta, \theta \xleftarrow{\$} \mathbb{F}_p$
2: $c' := h_T^{\theta}$
3: $\forall t \in \Gamma$, $c_t := \hat{g}_t^{\zeta}$
4: $z := h_T^{\zeta+\theta} = e_0(g_0, \hat{g}_0)^{\zeta+\theta}$
5: $c = m.z$
6: **return** $c_T := (\{c_t\}_{t\in\Gamma}, c, c', \Gamma)$

Decrypt(pk, $\mathsf{sk}_{\mathbb{S}}, c_T$):
Input: ((pk, $\mathsf{sk}_{\mathbb{S}}, c_T$)
Output: m' or $\perp$

1: **if** Γ satisfies $\mathbb{S}$ **then**
2: **for** $\rho(i) \in \Gamma$ **do**
3: computes σ_i such that $\vec{1} = \sum_{\rho(i)\in\Gamma} \sigma_i.M_i$
4:

$$z' := \prod_{t:=\rho(i)\in\Gamma} e_t(k_i, c_t)^{\sigma_i}.\, c'$$

5: **return** $m' := c.z'^{-1}$
6: **else**
7: **return** $\perp$

For the correctness of Isog-DBDH see in appendix B.1, and Isog-DLin see in appendix B.2

5 Large universe KP-ABE

Let $x_t := (x_{t,j})_{j\in[n]}$ be an attribute for any sub-universe id, where t is an element $\mathcal{U} := \{0,1\}^n$. This construction possesses a hierarchical structure for $t \in [d]$ and $j \in [n]$, representing two instantiations of a small universe. For n-bit attributes, we include n groups (elliptic curves) and encode the j-th group for $j \in [n]$. The IPG generates $2dn + 1$ pairing groups.

5.1 The Construction

Constructing a system with n bits attribute involves n groups of elliptic curves within the public parameters. Specifically, the j-th group is responsible for encoding the j-th bit, where $j \in [n]$. This approach is then extended to support a large universe of attributes.

$\underline{\mathsf{Setup}(\lambda, d)}$:
Input: (λ, d)
Output: $(\mathsf{pk}, \mathsf{msk})$
1: Choose $\mathsf{E}_0 : y^2 = x^3 + x$. ▷ a supersingular of elliptic curve
2: $g_0 \xleftarrow{\$} \mathbb{G}_0$, $\hat{g}_0 \xleftarrow{\$} \hat{G}_0$.
3: **for** $t \in [d]$ **do**,
4: **for** $j \in [n]$ **do**
5: **for** $\iota \in \{0, 1\}$ **do**
6: $(\mathsf{E}_{t,j,\iota}, \zeta_{t,j,\iota}) \leftarrow \mathsf{Isog}_{deg\phi,k}(\mathsf{E}_0)$.
7: Choose $h_T \xleftarrow{\$} \mathbb{G}_T$ such that $h_T := e_0(g_0, \hat{g}_0)$
8: $pk := \begin{cases} (\mathbb{G}_0, \hat{\mathbb{G}}_0, \hat{g}_0, e_0), \\ (\mathbb{G}_{t,j,\iota}, \hat{G}_{t,j,\iota}, \hat{g}_{t,j,\iota}, e_{t,j,\iota}), \\ h_T, \mathbb{G}_T) \end{cases}$
▷ In pk, $\iota \in [0, 1], t \in [d], j \in [n]$
9: $\mathsf{msk} := (\phi_{t,j,\iota})_{\iota \in [0,1]}^{t \in [d], j \in [n]}$
10: **return** $(\mathsf{pk}, \mathsf{msk})$.

$\underline{\mathsf{KeyGen}(\mathsf{pk}, msk, \mathbb{S} := (M, \rho))}$:
1: Choose a random vector $\vec{u} \xleftarrow{\$} \mathbb{F}_q^r$ such that $\vec{1}.\vec{u} = \sum_{i=1}^{r} u_i = s$ where s is secret to be shared and $(u_i) \xleftarrow{\$} \mathbb{F}_p^r$. There $\exists h'_0$ where $e_0(h'_0, \hat{g}_0) = h_T^{s^{-1}}$.
2: **for** i in [l] **do**
3: $s_i := M_i.u^T$ ▷ s_i are shares
4: Choose $\tau_i := \tau_{i,j}$ such that $s_i = \sum_{j=1}^{n} \tau_{i,j}$.
5: **if** $\rho(i) = (t, v_i := v_{i,j}) \in \{0, 1\}^n$ **then**
6: $k_{i,j} := \phi_{t,j,v_{i,j}}(h'_0)^{\tau_{i,j}}$
7: **return** $sk_{\mathbb{S}} := \{k_{i,j}\}_{i \in [l], j \in [n]}$.

$\underline{\mathsf{Encrypt}(\mathsf{pk}, m, \Gamma)}$:
1: $\zeta \xleftarrow{\$} \mathbb{F}_p$
2: **for** $(t, x_t := (x_{t,j}) \in \{0, 1\}^n) \in \Gamma$ **do**
3: $c_{t,j} := \hat{g}_{t,j,x_{t,j}}^{\zeta}$
4: $z := h_T^{\zeta} = e_0(g_0, \hat{g}_0)^{\zeta} \in \mathbb{G}_T$
5: $c := z.m$
6:
7: **return** $c_\Gamma := (\{c_{t,j}\}_{(t,.) \in \Gamma, j \in [n]}, c)$

$\underline{\mathsf{Decrypt}(\mathsf{pk}, \mathsf{sk}, c_T)}$:
1: **if** Γ satisfies $\mathbb{S} := \{(t, x_t)\}$ **then**
2: Computes $\{\sigma_i\}_{\rho(i) \in \Gamma}$ s.t $\vec{1} := \sum_{\rho(i) \in \Gamma} = \sigma_i.M_i$ $z' := \Pi_{\rho(i)=(t,v_{i,j}) \in \Gamma} \Big(\prod_{j=1}^{n} e_{t,j,v_{i,j}}(k_{i,j}, c_{t,j}) \Big)^{\sigma_i}$
3: $m' := c/z'$
4: **return** m'
5: **else**
6: **return** $\perp$

For the correctness see in appendix B.3

6 Security Analysis

Les preuves de sécurité de ta construction. This section will present the security game between an adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ and his challenger $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$. $\mathcal{B}_1$ will play the challenger in the phase for quantum $\mathcal{A}_1$ and $\mathcal{B}_2$ will play the challenger in the phase for classical adversary $\mathcal{A}_2$. **The Phase 1** is denoted **quantum phase against** $\mathcal{A}_1$ and **the phase 2** is denoted **classical phase against** $\mathcal{A}_2$. We note that ν_1 is the number query in the quantum phase and ν_2

is the number query in the classical phase. Let's begin with the KP-ABE of the small universe and terminate with the large universe.

Theorem 1. *The KP-ABE scheme is PH-PQ secure under the d-pIsog-DBDH assumption in the quantum random oracle model. For any adversary* $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$*, there exits* $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$*, such that for any security parameter* λ*:*

$$\mathrm{Adv}_{\mathcal{A}}^{KP-ABE,PH-PQ}(\lambda) \leq \mathrm{Adv}_{\mathcal{B}}^{d-pIsogDBDH}(\lambda)$$

Proof. Let start with :
Challenge phase for the quantum adversary $\mathcal{A}_1$

1. First $\mathcal{A}_1$ declares the challenge attributes Γ^* in which $\Gamma^* \notin \mathbb{S}_j$ for all j and provides Γ^* to $\mathcal{B}_1$.
2. $\mathcal{B}_1$ runs the Setup of KP-ABE and obtains $pp := (\mathbb{G}_t, \hat{G}_t, \hat{g}_t, e_t)_{t\in[d]}, h_T)$.
   $\mathcal{B}_2$ runs $(\mathbb{G}'_t, \hat{G'}_t, \hat{g}'_t, e'_t)_{t\notin\Gamma^* \, and \, t\in[d]}, h'_T) \xleftarrow{\$} \mathsf{SimGen}(\mathbb{G}_0, \hat{G}_0, \hat{g}_0, e_0)$.
   $pk := (\mathbb{G}_t, \hat{G}_t, \hat{g}_t, e_t,)_{t\in\Gamma^* \,\&\, t\in[d]}, h_T, (\mathbb{G}'_t, \hat{G'}_t, \hat{g}'_t, e'_t)_{t\notin\Gamma^*}, h'_T)$. Then $\mathcal{B}_1$ provides pk to $\mathcal{A}_1$.
3. $\mathcal{B}_1$ simulates as a challenger for $\mathcal{A}_1$ as:
   - Let $F(X)$ be a random degree ν polynomial $\mathcal{A}_1$, ie., $F(X) \leftarrow \bigoplus_{i=0}^{\nu} \mathbb{F}_p X^i$ with $\nu := 2\nu_1 + \nu_2$.
   - Let $s \xleftarrow{\$} \mathbb{F}_p^*$ such that $\exists h'_0 : e_0(h'_0, \hat{g}_0) = h_T^{1/s}$ $\tau := F(s)$. Hence a quantum random oracle query RO is answered by $h_0 := g_0^{\tau}$ where g_0 is from Setup.
   - A classical key generation query is answered as follows: $\mathcal{B}_1$ chooses a vector $\vec{u} \xleftarrow{\$} \mathbb{F}_p^r$ such that $\sum_{i=1}^{r} u_i = s$
     For $i \in [l]$ $s_i := M_i.\vec{u}^{\top}$, where M is the matrix of share of l rows and r columns.
     $\mathcal{B}_1$ returns $(k_i := g_{\rho(i)\in\Gamma^*}^{\tau s_i})_{i\in[l]}$ provides it to $\mathcal{A}_1$
4. $\mathcal{A}_1$ outputs $(state) \xleftarrow{\$} \mathcal{A}_1(pk)$ and sends it to $\mathcal{A}_2$ and B_2.

Challenge phase for the classical adversary $\mathcal{A}_2$

5. $\mathcal{B}_2$ gets $\mathcal{X}_b := (state', g_0^{\alpha}, (\hat{g}_0^{\beta})_{t\in[d]}, g_T^{\theta})$ where $\theta = \alpha\beta$ if $b = 0$ and otherwise $\theta \xleftarrow{\$} \mathbb{F}_p$. Then $\mathcal{B}_2$ sends $state$ to $\mathcal{A}_2$.
6. When a random oracle query is issued for access structure $\mathbb{S} := (M, \rho)$, $\mathcal{B}_2$ check:
   a- If Γ^* is not accepted by $\mathbb{S}_j$ $\mathcal{B}_2$ re-executes the KeyGen of step 3 in the challenge phase for the quantum adversary $\mathcal{A}_1$ with another $\mathbb{S}_{j+1}$.
   b- If Γ^* is accepted by $\mathbb{S}_j$, $\mathcal{B}_2$ generates a vector $\vec{v} \xleftarrow{\$} \mathbb{F}_p^r$.
   Takes $\vec{w} \xleftarrow{\$} \{\vec{w} \in \mathbb{F}_p^r | M_i.\vec{w}^{\top} = 0 \, if \rho(i) \in \Gamma^*, \mu := \vec{w}.\vec{1} \neq 0\}$
   Defines $\mathcal{X} = \frac{\alpha - \vec{v}.\vec{1}}{\mu}$, $\vec{u'} := \vec{v} + \mathcal{X}\vec{w}$, then implicitly $\alpha = \tau$.
   Thus let $\vec{u'} = \frac{\vec{u}}{\alpha}$ and $s_i := Mi.\vec{u}$ and $M_i.\vec{u'} = \alpha.s_i$. For $i \in [l], \mathcal{B}_2$ computes:

$$k_i := \begin{cases} g_t^{\eta_{1,i}} & if \rho(i) \in \Gamma^*, \\ \phi'_t((g_0^{\alpha})^{\eta_{2,i}}.g_0^{\eta_{3,i}}) & if \rho(i) \notin \Gamma^*. \end{cases}$$

where $\eta_{1,i} = M_i.\vec{v}$, $\eta_{2,i} = \frac{M_i.\vec{w}}{\mu}$ and $\eta_{3,i} = M_i.\vec{v} - \frac{M_i.\vec{w}.\vec{v}.\vec{1}}{\mu}$
If $\rho(i) \in \Gamma^*$, then $k_i = g_t^{\eta_{1,i}} = g_t^{M_i\vec{u}} = g_t^{\alpha s_i} = \phi_t(h_0)^{s_i}$.
If $\rho(i) \notin \Gamma^*$, then $k_i = \phi'_t((g_0^{\alpha})^{\eta_{2,i}+\eta_{3,i}} = (g'_t)^{M_i\vec{u}} = (g'_t)^{\alpha s_i} = \phi'_t(h'_0)^{s_i}$.
Then $\mathcal{B}_2$ sends the value $sk_{\mathbb{S}_j} = \{k_i\}_{i\in[l]}$ to $\mathcal{A}_2$.

7. $\mathcal{A}_2$ send two plaintexts m_0, m_1 such that $|m_0| = |m_1|$ to $\mathcal{B}_2$.
8. $\mathcal{B}_2$ encrypts plaintexts to obtain $c_T := \{(c_t)_{t\in\Gamma^*}, c\}$ such that $c_t := (\hat{g}_t^{\beta})_{t\in\Gamma^*}$ and $c := g_T^{\theta}.m_b$. $\mathcal{B}_2$ flips a random coin $b \in \{0,1\}$ and send c_T to $\mathcal{A}_2$.
9. After $\mathcal{A}_2$ issues a random oracle or a key query, $\mathcal{B}_2$ executes step 6.
10. Finally, $\mathcal{A}_2$ outputs b'.

Let ϵ be the advantage of $\mathcal{A}$.
$\Pr[b \neq b'|b' := 1] = \frac{1}{2}$
$\Pr[b = b'|b' := 1] = \frac{1}{2}$
$\Pr[b = b'|b := 0] = \frac{1}{2} + \epsilon$
$\frac{1}{2}\Pr[b = b'|b := 0] + \frac{1}{2} + (\Pr[b = b'|b' := 1]) - \frac{1}{2}$
$= \frac{1}{2} + (\frac{1}{2}\epsilon) + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2}$
$= \frac{\epsilon}{2}$
Conclusion: for any adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$, there exits $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$, such that for any security parameter λ:

$$\mathrm{Adv}_{\mathcal{A}}^{KP-ABE,PH-PQ}(\lambda) \leq \mathrm{Adv}_{\mathcal{B}}^{d-pIsogDBDH}(\lambda)$$

Theorem 2. *The KP-ABE scheme is PH-PQ secure under the d-pIsog-DLin assumption in the quantum random oracle model. For any adversary* $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$*, there exits* $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$*, such that for any security parameter* λ*:*

$$\mathrm{Adv}_{\mathcal{A}}^{KP-ABE,PH-PQ}(\lambda) \leq \mathrm{Adv}_{\mathcal{B}}^{\mathrm{d-pIsog-DLIN}}(\lambda)$$

Proof. Let start with :

Challenge phase for the quantum adversary $\mathcal{A}_1$

1. First $\mathcal{A}_1$ declares the challenge attributes Γ^* in which $\Gamma^* \notin \mathbb{S}_j$ for all j and provides Γ^* to $\mathcal{B}_1$.
2. $\mathcal{B}_1$ runs the Setup of KP-ABE and obtains $pp := (\mathbb{G}_t, \hat{G}_t, \hat{g}_t, e_t)_{t\in[d]}, h_T)$.
   $\mathcal{B}_2$ runs $(\mathbb{G}'_t, \hat{G}'_t, \hat{g}'_t, e'_t)_{t\notin\Gamma^* \, and \, t\in[d]}, h'_T) \xleftarrow{\$} \mathsf{SimGen}(\mathbb{G}_0, \hat{G}_0, \hat{g}_0, e_0)$.
   $pk := (\mathbb{G}_t, \hat{G}_t, \hat{g}_t, e_t)_{t\in\Gamma^* \,\&\, t\in[d]}, h_T, (\mathbb{G}'_t, \hat{G}'_t, \hat{g}'_t, e'_t)_{t\notin\Gamma^*}, h'_T)$. Then $\mathcal{B}_1$ provides pk to $\mathcal{A}_1$.
3. $\mathcal{B}_1$ simulates as a challenger for $\mathcal{A}_1$ as:
   - Let $F(X)$ be a random degree ν polynomial $\mathcal{A}_1$, ie., $F(X) \leftarrow \bigoplus_{i=0}^{\nu} \mathbb{F}_q X^i$ with $\nu := 2\nu_1 + \nu_2$.
   - Let $s \xleftarrow{\$} \mathbb{F}_q^*$ such that $\exists h'_0 : e_0(h'_0, \hat{g}_0) = h_T^{1/s}$ $\tau := F(s)$. Hence a quantum random oracle query RO is answered by $h_0 := g_0^{\tau}$ where g_0 is from setup.

– A classical key generation query is answered as follows: $\mathcal{B}_1$ chooses a vector $\vec{u} \xleftarrow{\$} \mathbb{F}_p^r$ such that $\sum_{i=1}^{r} u_i = s$
For $i \in [l]$ $s_i := M_i.\vec{u}^\top$, where M is the matrix of share of l rows and r columns.
$\mathcal{B}_1$ returns $(k_i := g_{\rho(i)\in\Gamma^*}^{\tau s_i})_{i\in[l]}$ provides it to $\mathcal{A}_1$

4. $\mathcal{A}_1$ outputs $(state) \xleftarrow{\$} \mathcal{A}_1(pk)$ and sends it to $\mathcal{A}_2$ and B_2.

Challenge phase for the classical adversary $\mathcal{A}_2$

5. $\mathcal{B}_2$ gets $\mathcal{X}_b := (\mathsf{state}', g_0^\alpha, g_0^\beta, (\hat{g}_t^\alpha, \hat{g}_t^\beta, \hat{g}_t^\gamma)_{t\in[d]}, g_T^\delta)$, where $\delta = \alpha\gamma + \beta\gamma$ if $b = 0$ and otherwise $\delta \xleftarrow{\$} \mathbb{F}_q$, $state' = (state, F(X))$.
6. When a random oracle query is issued for access structure $\mathbb{S} := (M, \rho)$, $\mathcal{B}_2$ check:
   a- If Γ^* is not accepted by $\mathbb{S}_j$, $\mathcal{B}_2$ re-executes the KeyGen of step 3 in the challenge phase for the quantum adversary $\mathcal{A}_1$ with another $\mathbb{S}_{j+1}$.
   b- If Γ^* is accepted by $\mathbb{S}_j$, $\mathcal{B}_2$ generates a vector $\vec{v} \xleftarrow{\$} \mathbb{F}_p^r$.
   Takes $\vec{w} \xleftarrow{\$} \{\vec{w} \in \mathbb{F}_p^r | M_i.\vec{w}^\top = 0 \, if \rho(i) \in \Gamma^*, \mu := \vec{w}.\vec{1} \neq 0\}$
   Defines $\mathcal{X} = \frac{\alpha+\beta-\vec{v}.\vec{1}}{\mu}$, $\vec{u'} := \vec{v} + \mathcal{X}\vec{w}$, then implicitly $\alpha + \beta = \tau$.
   Thus let $\vec{u'} = \frac{\vec{u}}{\alpha+\beta}$ and $s_i := M_i.\vec{u}$ and $M_i.\vec{u'} = (\alpha + \beta).s_i$. For $i \in [l], \mathcal{B}_2$ computes:
$$k_i := \begin{cases} g_t^{\eta_{1,i}} & if \rho(i) \in \Gamma^*, \\ \phi_t'((g_0^\alpha)^{\eta_{2,i}}.g_0^{\eta_{3,i}}) & if \rho(i) \notin \Gamma^*. \end{cases}$$
   where $\eta_{1,i} = M_i.\vec{v}$, $\eta_{2,i} = \frac{M_i.\vec{w}}{\mu}$ and $\eta_{3,i} = M_i.\vec{v} - \frac{M_i.\vec{w}.\vec{v}.\vec{1}}{\mu}$
   If $\rho(i) \in \Gamma^*$, then $k_i = g_t^{\eta_{1,i}} = g_t^{M_i\vec{u'}} = g_t^{(\alpha+\beta)s_i} = \phi_t(h_0)^{s_i}$.
   If $\rho(i) \notin \Gamma^*$, then $k_i = \phi_t'(g_0')^{(\alpha+\beta)\eta_{2,i}+\eta_{3,i}} = (g_t')^{M_i\vec{u'}} = (g_t')^{(\alpha+\beta)s_i} = \phi_t'(h_0')^{s_i}$.
   Then $\mathcal{B}_2$ sends the value $sk_{\mathbb{S}_j} = \{k_i\}_{i\in[l]}$ to $\mathcal{A}_2$.
7. $\mathcal{A}_2$ send two plaintexts m_0, m_1 such that $|m_0| = |m_1|$ to $\mathcal{B}_2$.
8. $\mathcal{B}_2$ encrypts plaintexts to obtain $c_T := \{(c_t)_{t\in\Gamma^*}, c, c'\}$ such that $c_t := (\hat{g}_t^\gamma)_{t\in\Gamma^*}$, $c' := g_T^\beta$ and $c := g_T^\delta.m_b$. $\mathcal{B}_2$ flips a random coin $b \in \{0, 1\}$ and send c_T to $\mathcal{A}_2$.
9. After $\mathcal{A}_2$ issues a random oracle or a key query, $\mathcal{B}_2$ executes step 6.
10. Finally, $\mathcal{A}_2$ outputs b'.

Conclusion: for any adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$, there exits $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$, such that for any security parameter λ:

$$\mathrm{Adv}_{\mathcal{A}}^{KP-ABE,PH-PQ}(\lambda) \leq \mathrm{Adv}_{\mathcal{B}}^{\mathrm{d-pIsog-DLIN}}(\lambda).$$

Theorem 3. *The KP-ABE scheme is PH-PQ under 2dn-pIsog-DBDH assumption in the random oracle. For any adversary $\mathcal{A}$, there an adversary $\mathcal{B}$ for the 2dn-pIsog-DBDH, such that for parameter security λ:*

$$\mathrm{Adv}_{\mathcal{A}}^{KP-ABE,PH-PQ}(\lambda) \leq \mathrm{Adv}_{\mathcal{B}}^{2dn-pIsogDBDH}(\lambda)$$

Proof. The proof of theorem 2 is similar to the theorem 1. Then the large universe can be demonstrated as follows:

Challenge phase for the quantum adversary $\mathcal{A}_1$

1. First $\mathcal{A}_1$ declares the challenge attributes Γ^* in which $\Gamma^* \notin \mathbb{S}_j$ for all j and provides Γ^* to $\mathcal{B}_1$.
2. $\mathcal{B}_2$ runs the Setup of **KP-ABE** with an input the security parameter λ and obtains:
   $\mathsf{pk} := ((\mathbb{G}_0, \hat{\mathbb{G}}_0, \hat{g}_0, \mathbb{G}_T, e_0), (\mathbb{G}_{t,j,\iota}, \hat{G}_{t,j,\iota}, \hat{g}_{t,j,\iota}, e_{t,j,\iota})_{\iota\in[0,1]}^{t\in[d],j\in[n]}, h_T, \mathbb{G}_T;$
   msk:$= (\phi_{t,j,\iota})_{\iota\in[0,1]}^{t\in[d],j\in[n]}$.
   $\mathcal{B}_2$ runs $((\mathbb{G}_{t,j,\iota}, \hat{G}_{t,j,\iota}, \hat{g}_{t,j,\iota}, e_{t,j,\iota})_{\iota\in[0,1]}^{t\in[d],j\in[n]}, h'_T) \xleftarrow{\$} \mathsf{SimGen}(\mathbb{G}_0, \hat{G}_0, \hat{g}_0, e_0)$.
3. $\mathcal{B}_1$ plays the role of challenger to $\mathcal{A}_1$ as :
   - Let $F(X)$ be a random degree ν polynomial $\mathcal{A}_1$, ie., $F(X) \leftarrow \bigoplus_{i=0}^{\nu} \mathbb{F}_p X^i$ with $\nu := 2\nu_1 + \nu_2$.
   - Let $s \xleftarrow{\$} \mathbb{F}_p^*$ such that $\tau := F(s)$. Hence a quantum random oracle query RO is answered by $h_0 := g_0^{\tau}$ where g_0 is from Setup.
   - A classical key generation query is answered as follows: $\mathcal{B}_1$ choose a random vector $\vec{u} \xleftarrow{\$} \mathbb{F}_p^r$ such that $\sum_{i=1}^{r} u_i = s$. For $i \in [l] s_i := M_i.u^T$ such that there exist a random vector $\vec{\tau} = (\tau_{i,j})$ and $s_i := \sum_{j=1}^{n} \tau_{i,j}$.
     If $\rho(i) = (t, v_i = (v_{i,j}) \in \{0,1\}^n$, $(k_{i,j} := g_{\rho(i)\in\Gamma^*})_{i\in[l]}^{j\in[n]}$.
4. $\mathcal{A}_1$ outputs $(state) \xleftarrow{\$} \mathcal{A}_1(pk)$ and sends it to $\mathcal{A}_2$ and B_2.

Challenge phase for the classical adversary $\mathcal{A}_2$

5. $\mathcal{B}_2$ gets $\mathcal{X}_b := (state', g_0^{\alpha}, (\hat{g}_0^{\beta})_{t\in[d]}, g_T^{\theta})$ where $\theta = \alpha\beta$ if $b = 0$ and otherwise $\theta \xleftarrow{\$} \mathbb{F}_p$. Then $\mathcal{B}_2$ sends $state$ to $\mathcal{A}_2$.
6. When a random oracle query is issued for access structure $\mathbb{S} := (M, \rho)$, $\mathcal{B}_2$ check:
   a- If Γ^* is not accepted by $\mathbb{S}_j$ $\mathcal{B}_2$ re-executes the KeyGen of step 3 in the challenge phase for the quantum adversary $\mathcal{A}_1$ with another $\mathbb{S}_{j+1}$.
   b- If Γ^* is accepted by $\mathbb{S}_j$, $\mathcal{B}_2$ generates a vector $\vec{v} \xleftarrow{\$} \mathbb{F}_p^r$.
   $\mathcal{B}_2$ generates $\vec{v_i} \xleftarrow{\$} \mathbb{F}_p^r$ Takes $\vec{w_i} \xleftarrow{\$} \{\vec{w_i} \in \mathbb{F}_p^r | M_i.\vec{w_i}^{\top} = 0 \, if \rho(i) \in \Gamma^*, \mu_i := \vec{w_i}.\vec{1} \neq 0\}$
   Defines $\mathcal{X} = \frac{\alpha - \vec{v_{i_i}}.\vec{1}}{\mu_i}$, $\vec{u_i'} := \vec{v}_i + \mathcal{X}\vec{w}_i$, then implicitly $\alpha = \tau_i$.
   Thus let $\vec{u'}_i = \frac{\vec{u}_i}{\alpha}$ and $s_i := M_i.\vec{u}_i$ and $M_i.\vec{u'}_i = \alpha.s_i$.
   For $i \in [l]$,
   - For $j \in [n] \mathcal{B}_2$ computes:

$$k_{i,j} := \begin{cases} g_t^{\eta_{i,j}} & if \rho(i) \in \Gamma^*, \\ \phi'_t((g_0^{\alpha})^{\eta'_{i,j}}.g_0^{\eta"_{i,j}}) & if \rho(i) \notin \Gamma^*. \end{cases}$$

   where $\eta_{i,j} = M_i.\vec{v}_i$, $\eta'_{i,j} = \frac{M_i.\vec{w}'}{\mu_i}$ and $\eta"_{i,j} = M_i.\vec{v}_i - \frac{M_i.\vec{w}_i.\vec{v}_i.\vec{1}}{\mu}$
   If $\rho(i) \in \Gamma^*$, then $k_{i,j} = g_t^{\eta_{i,j}} = g_t^{M_i\vec{u'}_i} = g_t^{\alpha s_i} = \phi_{t,j,v_{i,j}}(h_0)^{s_i}$.

If $\rho(i) \notin \Gamma^*$, then $k_{i,j} = \phi'_t((g_0^{\alpha})^{\eta'_{i,j}+\eta"_{i,j}} = (g'_t)^{M_i\vec{u'}_i} = (g'_t)^{\alpha s_i} = \phi'_{t,j,v_{i,j}}(h'_0)^{s_i}$.

Then $\mathcal{B}_2$ sends the value $sk_{\mathbb{S}_j} = \{k_{i,j}\}_{i\in[l]}^{j\in[n]}$ to $\mathcal{A}_2$.

item $\mathcal{A}_2$ send two plaintexts m_0, m_1 such that $|m_0| = |m_1|$ to $\mathcal{B}_2$.

7. $\mathcal{B}_2$ encrypts plaintexts to obtain $c_T := \{(c_{t,j})_{(t,.)\in\Gamma^*,j\in[n]}, c\}$ such that $c_t := (\hat{g}^{\beta})_{t,j,v_{i,j}\in\Gamma^*}$ and $c := g_T^{\theta}.m_b$. $\mathcal{B}_2$ flips a random coin $b \in \{0,1\}$ and send c_T to $\mathcal{A}_2$.
8. After $\mathcal{A}_2$ issues a random oracle or a key query, $\mathcal{B}_2$ executes step 6.
9. Finally, $\mathcal{A}_2$ outputs b'.

Therfore, for any adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$, there exits $\mathcal{B} := (\mathcal{B}_1, \mathcal{B}_2)$, such that for any security parameter λ:

$$\mathrm{Adv}_{\mathcal{A}}^{KP-ABE,PH-PQ}(\lambda) \leq \mathrm{Adv}_{\mathcal{B}}^{2dn-IsogDBDH}(\lambda)$$

Conclusion

We encountered several challenges while exploring attribute-based encryption (ABE) based on isogeny. The primary difficulty is that the current isogeny structure does not support an ABE scheme with quantum-resistant ciphertext. We have identified a security concept called payload hiding against quantum attacks and constructed a key-policy attribute-based encryption (KP-ABE) scheme using an isogeny pairing group. This involves introducing a new assumption, Isog-DLin, within a small universe. Future research will investigate constructing ciphertext-policy attribute-based encryption (CP-ABE) schemes using isogeny pairing groups and group actions based on oriented supersingular curves, such as SCALLOP or SCALLOP HD, for improved computational efficiency.

Acknowledgement. This paper results from an exploratory internship financed by Institut de Recherche Technologique SystemX (IRT-SystemX).

References

1. Affum, E., Zhang, X., Wang, X., Ansuura, J.B.: Efficient lattice cp-abe ac scheme supporting reduced-obdd structure for ccn/ndn. Symmetry **12**(1) (2020). https://doi.org/10.3390/sym12010166, https://www.mdpi.com/2073-8994/12/1/166
2. Affum, E., Zhang, X., Wang, X., Ansuura, J.B.: Efficient lattice cp-abe ac scheme supporting reduced-obdd structure for ccn/ndn. Symmetry **12**(1), 166 (2020)
3. Beimel, A., et al.: Secure schemes for secret sharing and key distribution. Thesis (1996)
4. 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)
5. Boyen, X.: Attribute-based functional encryption on lattices. In: Theory of cryptography conference. pp. 122–142. Springer (2013)

6. Castryck, W., Decru, T.: Csidh on the surface. In: International Conference on Post-Quantum Cryptography. pp. 111–129. Springer (2020)
7. Castryck, W., Decru, T.: An efficient key recovery attack on sidh. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 423–447. Springer (2023)
8. Castryck, W., Lange, T., Martindale, C., Panny, L., Renes, J.: Csidh: an efficient post-quantum commutative group action. In: Advances in Cryptology–ASIACRYPT 2018: 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2–6, 2018, Proceedings, Part III 24. pp. 395–427. Springer (2018)
9. Chen, L., Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R.A., Smith-Tone, D.: Report on post-quantum cryptography, vol. 12. US Department of Commerce, National Institute of Standards and Technology ... (2016)
10. Chen, Y.: Quantum algorithms for lattice problems (2024), https://eprint.iacr.org/2024/555, https://eprint.iacr.org/2024/555
11. Couveignes, J.M.: Hard homogeneous spaces. Cryptology ePrint Archive (2006)
12. Dai, W., Doröz, Y., Polyakov, Y., Rohloff, K., Sajjadpour, H., Savaş, E., Sunar, B.: Implementation and evaluation of a lattice-based key-policy abe scheme. IEEE Transactions on Information Forensics and Security **13**(5), 1169–1184 (2017)
13. Fugeng Zeng, C.X.: A novel model for lattice-based authorized searchable encryption with special keyword. Mathematical Problems in Engineering **Article ID 314621** (2015). https://doi.org/https://doi.org/10.1155/2015/314621
14. Fun, T.S., Samsudin, A.: Attribute based encryption—a data centric approach for securing internet of things (iot). Advanced Science Letters **23**(5), 4219–4223 (2017)
15. Galbraith, S.D., Petit, C., Shani, B., Ti, Y.B.: On the security of supersingular isogeny cryptosystems. In: Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22. pp. 63–91. Springer (2016)
16. Gavriloaie, R., Nejdl, W., Olmedilla, D., Seamons, K.E., Winslett, M.: No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web. In: European Semantic Web Symposium. pp. 342–356. Springer (2004)
17. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996)
18. Harney, H., Colgrove, A., McDaniel, P.: Principles of policy in secure groups. In: 8th Symposium on Network and Distributed System Security, NDSS 2001. The Internet Society (2001)
19. Hou, J., Peng, C., Tan, W., Ding, H.: Quantum-resistant multi-feature attribute-based proxy re-encryption scheme for cloud services. CMES-Computer Modeling in Engineering & Sciences **138**(1), 917–938 (2024)
20. Jao, D., De Feo, L.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29–December 2, 2011. Proceedings 4. pp. 19–34. Springer (2011)
21. Kang, M.H., Park, J.S., Froscher, J.N.: Access control mechanisms for inter-organizational workflow. In: Proceedings of the sixth ACM symposium on Access control models and technologies. pp. 66–74 (2001)

22. Khajouei-Nejad, S., Javadi, H.H.S., Jabbehdari, S., Moattar, S.M.H.: Reducing the computational complexity of fuzzy identity-based encryption from lattice. Cryptology ePrint Archive (2024)
23. Koshiba, T., Takashima, K.: Pairing cryptography meets isogeny: A new framework of isogenous pairing groups. Cryptology ePrint Archive (2016)
24. Koshiba, T., Takashima, K.: New assumptions on isogenous pairing groups with applications to attribute-based encryption. In: Information Security and Cryptology–ICISC 2018: 21st International Conference, Seoul, South Korea, November 28–30, 2018, Revised Selected Papers 21. pp. 3–19. Springer (2019)
25. Liu, Z., Jiang, Z.L., Wang, X., Wu, Y., Yiu, S.M.: Multi-authority ciphertext policy attribute-based encryption scheme on ideal lattices. In: 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/-SustainCom). pp. 1003–1008. IEEE (2018)
26. Luo, F., Al-Kuwari, S., Wang, F., Chen, K.: Attribute-based proxy re-encryption from standard lattices. Theoretical Computer Science **865**, 52–62 (2021)
27. Nikolaenko, V.: Studies in Secure Computation: Post-Quantum, Attribute-Based and Multi-Party. Ph.D. thesis, Stanford University (2017)
28. Pal, T., Dutta, R.: Attribute-based access control for inner product functional encryption from lwe. In: International Conference on Cryptology and Information Security in Latin America. pp. 127–148. Springer (2021)
29. Qian, X., Wu, W.: An efficient ciphertext policy attribute-based encryption scheme from lattices and its implementation. In: 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS). pp. 732–742. IEEE (2021)
30. Rostovtsev, A., Stolbunov, A.: Public-key cryptosystem based on isogenies. Cryptology ePrint Archive (2006)
31. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005. Proceedings 24. pp. 457–473. Springer (2005)
32. Shekhawat, H., Gupta, D.S.: A survey on lattice-based security and authentication schemes for smart-grid networks in the post-quantum era. Concurrency and Computation: Practice and Experience p. e8080 (2024)
33. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)
34. Silverman, J.H.: The arithmetic of elliptic curves, vol. 106. Springer (2009)
35. Soo Fun, T., Samsudin, A.: Lattice ciphertext-policy attribute-based encryption from ring-lwe. In: 2015 International Symposium on Technology Management and Emerging Technologies (ISTMET). pp. 258–262 (2015). https://doi.org/10.1109/ISTMET.2015.7359040
36. TAN), S.F., SAMSUDIN, A.: Ciphertext policy-attribute based homomorphic encryption (cp-abher-lwe) scheme: A fine-grained access control on outsourced cloud data computation. Journal of Information Science and Engineering **33**(3), 675–694 (May 2017). https://doi.org/10.6688/JISE.2017.33.3.5
37. Tsabary, R.: Fully secure attribute-based encryption for t-cnf from lwe. In: Annual International Cryptology Conference. pp. 62–85. Springer (2019)
38. Varri, U.S., Pasupuleti, S.K., Kadambari, K.: Cp-absel: Ciphertext-policy attribute-based searchable encryption from lattice in cloud storage. Peer-to-Peer Networking and Applications **14**, 1290–1302 (2021)

39. Vélu, J.: Isogénies entre courbes elliptiques. Comptes-Rendus de l'Académie des Sciences **273**, 238–241 (1971)
40. Waters, B.: Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In: Annual international cryptology conference. pp. 619–636. Springer (2009)
41. Yang, K., Wu, G., Dong, C., Fu, X., Li, F., Wu, T.: Attribute based encryption with efficient revocation from lattices. Int. J. Netw. Secur. **22**(1), 161–170 (2020)
42. Yu, J., Yang, C., Tang, Y., Yan, X.: Attribute-based encryption scheme supporting tree-access structure on ideal lattices. In: Cloud Computing and Security: 4th International Conference, ICCCS 2018, Haikou, China, June 8–10, 2018, Revised Selected Papers, Part III 4. pp. 519–527. Springer (2018)
43. Zhang, J., Zhang, Z.: A ciphertext policy attribute-based encryption scheme without pairings. In: Wu, C.K., Yung, M., Lin, D. (eds.) Information Security and Cryptology. pp. 324–340. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
44. Zhang, J., Zhang, Z., Ge, A.: Ciphertext policy attribute-based encryption from lattices. In: Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security. p. 16–17. ASIACCS '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2414456.2414464, https://doi.org/10.1145/2414456.2414464
45. Zhao, J., Gao, H.: Lsss matrix-based attribute-based encryption on lattices. In: 2017 13th International Conference on Computational Intelligence and Security (CIS). pp. 253–257. IEEE (2017)

A Isogeny Sequence

1. **Initialization**
   - First, we refer to CSIDH 512 and Csi-Fish. Let p be a large prime where $p+1 = 4 \cdot 587 \prod_{i=1}^{n=73} l_i$ (all l_i are odd primes and $\left(\frac{-p}{l_1}\right) = -1$). Choose a generator $\mathfrak{g}^3 := \mathfrak{l}_1^3 = \langle 3, \pi - 1 \rangle^3$.
   - Let $\mathsf{E}(\mathbb{F}_p)$ be the set of supersingular elliptic curves defined over $\mathbb{F}_p$ ($\#\mathsf{E}(\mathbb{F}_p) = p+1$), and let $e_N(.,.)$ be the Weil pairing on $\mathsf{E}[N]$. $\mathsf{E}(\mathbb{F}_p)$ contains exactly one cyclic subgroup $\mathbb{G} := \mathsf{E}[l^k] \cap \mathsf{E}(\mathbb{F}_p)$ of order l^k.
   - Let $u \in \mathbb{F}_{p^2} \setminus \mathbb{F}_p$ such that $u^2 \in \mathbb{F}_p$. We define the map

$$\begin{aligned} v : \mathsf{E} &\to \hat{\mathsf{E}} \\ (x, y) &\mapsto (u^2 x, u^3 y) \end{aligned}$$

     to a quadratic twist $\hat{\mathsf{E}}$ of E, i.e., to a curve that is isomorphic to E over $\mathbb{F}_{p^2}$ and $\#\hat{\mathsf{E}}(\mathbb{F}_p) = p+1$.
   - Let $\mathsf{E}_0 : y^2 = x^3 + x$ be a supersingular elliptic curve defined over $\mathbb{F}_p$.

   Hence, we choose $\mathfrak{l}_1$ such that $l_1 = 3$, and all torsion points are defined over $\mathbb{F}_{p^2}$.
   We note that $3^k \mid p+1$.
2. **Isogeny Sequence**
   - Second, we can compute an isogeny of degree 3^k with Algorithm 1 [23]. Then, we have a curve E and a point generator $\zeta = R$.

– This allows us to compute two cyclic subgroups: $\hat{\mathbb{G}} := \hat{\mathsf{E}}[l^k] \cap \hat{\mathsf{E}}(\mathbb{F}_p)$ of order 3^k, and thus $\mathbb{G} := v^{-1}(\hat{\mathbb{G}})$.

We describe it in the following algorithm:

Algorihm 1: $\mathrm{Isog}_{l,k}$

Input: A supersingular elliptic curve E_0
Output: An isogenous E, a kernel ζ that is the trapdoor , and two cyclic groups $(\mathbb{G}, \hat{G})$ ▷ All instantiation is based on Csi-Fish

1: generates a random point $R \in \mathsf{E}[3^k]$, then $R_0 := R$
2: **for** $0 \leq i \leq k$ **do** $\mathsf{E}_{i+1} := \mathsf{E}_i/\mathsf{E}[3^{k-i-1}]$, $\psi : \mathsf{E}_i \to \mathsf{E}_{i+1}$, and $R_{i+1} := \psi(R_i)$ by Vélu's formula
3: Do composition $\phi := \psi_{k-1} \circ \psi_0 : \mathsf{E} \to \mathsf{E}_k = E/E < R >$.
4: $\mathsf{E} := \mathsf{E}_k$ and $\zeta := R$
5: $\hat{G} := \hat{\mathsf{E}}[l^k] \cap \hat{\mathsf{E}}(\mathbb{F}_p)$
6: $G := v^{-1}(\hat{\mathbb{G}})$
7: **return** $(\mathsf{E}, \zeta, \mathbb{G}, \hat{G})$

Note that for curves E, E' such that $\phi : \mathsf{E} \to \mathsf{E}'$ and cyclic groups $\mathbb{G}, \hat{\mathbb{G}}$ for E respectively $\mathbb{G}', \hat{\mathbb{G}}'$ for E':

$$e(g, \hat{g}) = e'(g', \hat{g'}) = e(\phi(g), \hat{g'})$$

A.1 Instantiation of Isogeny Pairing Group (IPG)

The following algorithms are an instantiation of an IPG generator.

Algorithm 2: $\mathsf{Gen}^{\mathsf{IPG}}(\lambda, d)$ **[23]**

Input: (λ, d)
Output: $(\mathsf{pk}^{IPG}, \mathsf{sk}^{IPG})$

1: Generate a supersingular elliptic curve $\mathsf{E}_0 := y^2 = x^3 + x$ over $\mathbb{F}_p$ with a cardinality 3^k ▷ $(\mathbb{G}, \hat{\mathbb{G}}, \mathbb{G}_T, e_0)$ is an asymmetric pairing group of order r from E_0 such that e_0 is a Weil pairing, $g_0 \xleftarrow{\$} \mathbb{G}_0$ and $\hat{g}_0 \xleftarrow{\$} \hat{G}_0$ following section A (Initialization)
2: **for** $t \in [d]$ **do**
3: $(\mathsf{E}_t, \zeta_t) \xleftarrow{\$} \mathrm{Isog}_{l,k}(\mathsf{E}_0), \phi_t := \phi_{\zeta_t}$, $\mathbb{G}_t := \phi_t(\mathbb{G}_0)$, $\hat{\mathbb{G}}_t := \phi_t(\hat{G}_0)$
4: choose a random h_T such that $h_T := e_0(h_0, \hat{g}_0)$ ▷ $h_0 \in \mathbb{G}_0$ and $\hat{g}_0 \in \hat{\mathbb{G}}_0$
5: **return** $\mathsf{pk}^{IPG} := ((\mathbb{G}_t, \hat{\mathbb{G}}_t, g_t, \hat{g}_t, e_t)_{t\in[d]}, h_T, \mathbb{G}_T), \mathsf{sk}^{IPG} := (\zeta_t)_{t\in[d]}$ for $(\phi_t)_{t\in[d]})$

Algorihm 3: SimGen [23]

Input: $(\mathbb{G}_0, \hat{G}_0, g, \hat{g}_0, e_0)$:
Output: $(\mathbb{G}, \hat{G}, g, \hat{g}, e, \zeta \text{ for } \phi)$

1: $(\mathsf{E}_t, \zeta_t) \xleftarrow{\$} \text{Isog}_{l,k}(\mathsf{E}_0)$
2: $\phi := \phi_\zeta$, , $g := \phi(g_0)$, $\mathbb{G} := \phi(\mathbb{G}_0)$, $\hat{\mathbb{G}}_0 := \phi(\hat{\mathbb{G}}_0)$ ▷
$e(h, \hat{h}) := e_{Weil}(h, \hat{h})$ for any $h \in \mathbb{G}$, $\hat{h} \in \hat{\mathbb{G}}$, where e_{Weil} is the Weil pairing on E
3: **return** $(\mathbb{G}, \hat{G}, g, \hat{g}, e, \zeta \text{ for } \phi)$

B Correctness

The following lines show the correctness of the small and large universe of KP-ABE.

B.1 Correctness of KP-ABE under Isog-DBDH assumption

The following lines show the correctness of the small universe construction under Isog-DBDH assumption.

Correctness:

If $\mathbb{S}$ accepts Γ then

$$z' := \prod_{t:=\rho(i)\in\Gamma} e_t(k_i, c_t)^{\sigma_i}$$
$$z' := \prod_{t:=\rho(i)\in\Gamma} e_t(\phi_t(h'_0)^{s_i}, \hat{g}_t^{\zeta})^{\sigma_i}$$
$$z' := \prod_{t:=\rho(i)\in\Gamma} e_t(\phi_t(h'_0)^{s_i}, \phi_t(\hat{g}_0)^{\zeta})^{\sigma_i}$$
$$z' := \prod_{t:=\rho(i)\in\Gamma} e_t(\phi_t(h'_0), \phi_t(\hat{g}_0))^{\zeta s_i \sigma_i}$$
$$z' := \prod_{\rho(i)\in\Gamma} e_0(h'_0, \hat{g}_0)^{\zeta s_i \sigma_i}$$
$$z' := e_0(h'_0, \hat{g}_0)^{\zeta \sum_{\rho(i)\in\Gamma} s_i \sigma_i}$$
$$z' := e_0(h'_0, \hat{g}_0)^{\zeta s}$$
$$z' := h_T^{\zeta s^{-1} s}$$
$$z' = z$$

B.2 Correctness of the small universe of KP-ABE under Isog-DLin assumption

In the algorithm Encrypt, $z := h_T^{\zeta} = e_0(h_0, \hat{g}_0)^{\zeta+\theta}$ and $c = m.z$.

Correctness:

If $\mathbb{S}$ accepts Γ then

$$z' := \prod_{t:=\rho(i)\in\Gamma} e_t(k_i, c_t)^{\sigma_i}.c'$$
$$z' := \prod_{t:=\rho(i)\in\Gamma} e_t(\phi_t(h'_0)^{s_i}, \hat{g}_t^{\zeta})^{\sigma_i}.c'$$
$$z' := \prod_{t:=\rho(i)\in\Gamma} e_t(\phi_t(h'_0)^{s_i}, \phi_t(\hat{g}_0)^{\zeta})^{\sigma_i}.c'$$
$$z' := \prod_{t:=\rho(i)\in\Gamma} e_t(\phi_t(h'_0), \phi_t(\hat{g}_0))^{\zeta s_i \sigma_i}.c'$$
$$z' := \prod_{\rho(i)\in\Gamma} e_0(h'_0, \hat{g}_0)^{\zeta s_i \sigma_i}.c'$$
$$z' := e_0(h'_0, \hat{g}_0)^{\zeta \sum_{\rho(i)\in\Gamma} s_i \sigma_i}.c'$$
$$z' := e_0(h'_0, \hat{g}_0)^{\zeta s}.c'$$
$$z' := h_T^{\zeta s^{-1} s}.h_T^{\theta}$$
$$z' := h_T^{\zeta+\theta}$$
$$z' = z$$

B.3 Large universe of KP-ABE

We know that in the algorithm of Encrypt, $z := h_T^{\zeta} = e_0(h_0, \hat{g}_0)^{\zeta} \in \mathbb{G}_T$ and $c := z.m$. The correctness is :

Correctness:

If $\mathbb{S}$ accepts Γ

$$z' = \prod_{\rho(i)=(t,v_{i,j})\in\Gamma} \left(\prod_{j=1}^{n} e_{t,j,v_{i,j}}(k_{i,j}, c_{t,j}) \right)^{\sigma_i}$$
$$z' := \prod_{\rho(i)=(t,v_{i,j})\in\Gamma} \left(\prod_{j=1}^{n} e_{t,j,v_{i,j}}(\phi_{t,j,v_{i,j}}(h'_0)^{\tau_{i,j}}, \hat{g}_{t,j,x_{t,j}}^{\zeta}) \right)^{\sigma_i}$$
$$z' := \prod_{\rho(i)=(t,v_{i,j})\in\Gamma} \left(\prod_{j=1}^{n} e_{t,j,v_{i,j}}(\phi_{t,j,v_{i,j}}(h'_0)^{\tau_{i,j}}, \phi_{t,j,v_{i,j}}(\hat{g}_0)^{\zeta}) \right)^{\sigma_i}$$
$$z' := \prod_{\rho(i)=(t,.)\in\Gamma} \left(\prod_{j=1}^{n} e_0(h'_0, \hat{g}_0)^{\tau_{i,j}\zeta} \right)^{\sigma_i}$$
$$z' := \prod_{\rho(i)=(t,.)\in\Gamma} \left(e_0(h'_0, \hat{g}_0)^{s_i\zeta} \right)^{\sigma_i}$$
$$z' := e_0(h'_0, \hat{g}_0)^{\zeta.s.s^{-1}}$$
$$z' = z$$