Ciphertext Policy Attribute Based Encryption for Arithmetic Circuits

Mahdi Mahdavi Oliaee, Zahra Ahmadian*

Abstract

Applying access structure to encrypted sensitive data is one of the challenges in communication networks and cloud computing. Various methods have been proposed to achieve this goal, one of the most interesting of which is Attribute-Based Encryption (ABE). In this method, the access structure, which is defined as a policy, can be applied to the key or ciphertext. Thus, if the policy is applied to the key, it is called Key Policy Attribute-Based Encryption (KP-ABE), and on the other hand, if it is applied to the ciphertext, it is called Ciphertext Policy Attribute-Based Encryption (CP-ABE). Since in the KP-ABE, the policy is selected once by a trusted entity and is fixed, they are not suitable for applications where the policy needs to change constantly. But on the other hand, In CP-ABE when the policy is selected by the sender and changed for each message, this problem is solved. Furthermore, the access structure should present a strong fine-grained access control. The arithmetic access structure can supply fine-grained access structures stronger than Boolean access structures.

We present the first CP-ABE scheme with an arithmetic circuit access policy based on the multilinear maps. First, we outline a basic design and then two improved versions of this scheme, with or without the property of hidden attributes, are introduced. We also define the concept of Hidden Result Attribute Based Encryption (HR-ABE) which means that the result of the arithmetic function will not be revealed to the users.

We define a new hardness assumption, called the $(k-1)$-Distance Decisional Diffie-Hellman assumption, which is at least as hard as the k-multilinear decisional Diffie-Hellman assumption. Under this assumption, we prove the adaptive security of the proposed scheme.

*The authors are with the Electrical Engineering Department, Shahid Beheshti University, Tehran, Iran. e-mail: m_mahdavioliaee@sbu.ac.ir, z_ahmadian@sbu.ac.ir
The corresponding author is Zahra Ahmadian.

Keywords: Ciphertext Policy Attribute Based Encryption (CP-ABE), Arithmetic circuit, Multilinear map, Adaptive security, Hidden attributes, Hidden Result.

1. Introduction

Nowadays, there is a considerable demand for fine-grained data sharing in cloud based communication systems, where access to the data is supposed to be limited to some specific eligible users. This type of data sharing requires flexible and dynamic access control over a service provider which is not necessarily trusted enough. Based on the traditional public key encryption solutions, the sender must identify all the potential qualified users and encrypt the message for each of them separately, which is an extremely inefficient solution. Attribute Based Encryption (ABE) addresses this demand by providing a dynamic access control based on the user's set of attributes. The access structure, which is itself protected by encryption, can be embedded in either the key (KP-ABE) or the ciphertext (CP-ABE). The flexibility of ABE makes it applicable to many different aspects of recent technologies, such as Internet of Things [2, 22], personal healthcare records [27, 24], Internet of Energy [25] and vehicular networks [10].

Related work. The concept of Attribute Based Encryption was first invented by Sahai and Waters [23], though under the title of fuzzy Identity Based Encryption (fuzzy IBE). In their scheme, each user has a set of attributes and a set of secret keys associated with these attributes. The message is encrypted by the sender based on the attributes and if the intersection of the sender and receiver attribute sets is greater than a predefined threshold value, the message can be decrypted by the receiver.

Goyal et al. [14] defined the concept of Key Policy Attribute Based Encryption (KP-ABE) and proposed the first KP-ABE scheme. In this type of ABE scheme, the ciphertext is labeled with a set of attributes, and the user's secret key is associated with an access structure. The ciphertext is decryptable only by the users whose secret key access structure is satisfied by the set of attributes attached to the ciphertext. Contrary to KP-ABE, Goyal et al. also introduced the concept of Ciphertext Policy Attribute Based Encryption (CP-ABE), though they did not propose a scheme with such a property.

In 2007, the first CP-ABE scheme was proposed by Bethencourt et al. [6]. In this type of ABE, the ciphertext is constructed according to the access structure and the secret keys of the receiver are constructed according to the user's attributes. The set of attributes of the decryptor in CP-ABE must satisfy the access structure defined in the ciphertext. Due to the possibility of choosing the access structure by the sender, this scheme is more flexible than KP-ABE.

Bethencourt proved the security of his scheme in the generic group model. Waters in [26] proposed a CP-ABE scheme and demonstrated the security of his scheme under standard assumptions. All of these schemes support the monotone circuit access structures. Ostrovsky et al. [21] presented the first schemes for non-monotone circuits. Green et al. [15] proposed the idea of outsourcing the heavy computations to the cloud, to reduce the computational overhead for the users. Some other articles in this research area focus on improving the efficiency, security, and size of the ciphertext and keys [4],[19], [29], [11] and [18].
One challenge in this area is the problem of revocation. Some schemes, like [32] and [16], focus on resolving this problem. Chase in [9] proposed the multi-authority ABE as a solution for the key escrow problem. In [3], Attrapondong and Imai present the Dual Policy ABE, which is a kind of ABE with simultaneous key and ciphertext policies. In [31, 20], the Hierarchical Attribute Based Encryption (HABE) was presented. In HABE, the user possessing an attribute with a higher level can decrypt the messages encrypted for that with lower level ones.
Garg et al. in [12] presented a backtracking attack for pairing-based ABE with circuits with fan-out bigger than one. Garg presented KP-ABE for all circuits using multilinear maps, though the underlying assumptions for proving its security are related to multilinear maps. However, his scheme works for any circuits with arbitrary fanout. The other scheme that supports arbitrary fanout was proposed in [13].

All the above schemes are constructed based on the bilinear pairing and their security relies on pairing-related hard problems. Therefore, they can not be regarded as post quantum ABE schemes. Contrary to pairing based ABE schemes, lattice based ABE schemes are proposed, where security relies on the Learning With Error (LWE) assumption. Agrawal et al. [1] presented the Fuzzy ABE based on lattice for the first time. Boyen et al. [8] and Zhang et al. [30] presented the first lattice-based KP-ABE and CP-ABE, respectively. Gorbunov et al. [13] presented the lattice based KP-ABE that works for any boolean. The first work which supports the arithmetic circuit as the access structure is Boneh's scheme [7], where a fully key homomorphic encryption for constructing KP-ABE is proposed. In this scheme, addition and multiplication gates are used instead of the conventional AND and OR gates, which is a more general approach than the boolean access structures.
Some schemes have the property of hiding the attribute vector or access policy in the ciphertext. This property is called Predicate Encryption (PE) [17]. Such ABE schemes are called policy hidden ABE [5, 28].

Motivation. All the above schemes, except Boneh et.al. [7] which is designed for arithmetic functions, are constructed for boolean functions. Arithmetic functions can achieve stronger fine-grained access control than boolean functions. Furthermore, it is possible to generate the boolean access policy from the arithmetic function [7]. So, arithmetic functions as access policy is more general and more flexible than the boolean ones. So, this work focuses on new ABE schemes with arithmetic access functions, aiming to mitigate some limitations of [7].

Our contribution. In this paper, we propose the first CP-ABE scheme for arithmetic functions with arbitrary results. The proposed scheme is designed based on the multilinear map. We introduce the new concept of hidden result ABE, which means that the result of the arithmetic function remains unknown to the user.

The proposed scheme is described in three variants. A basic scheme is first introduced by which the platform of our idea is demonstrated. In this scheme, the result and attribute vector is hidden and it covers simple arithmetic functions. Then, an improved version, supporting a general arithmetic function is proposed in which the attribute vector, as well as the result value, are unknown to the users. Compared to [7], which is the only existing ABE work for arithmetic circuits, the proposed scheme has significant advantages. Our proposed schemes are CP-ABE with adaptive security. The result can take any arbitrary value. It supports the exponentiation gate and does not have any constraint over the attribute values. None of the above features are supported by Boneh's scheme [7]. However, that scheme is a lattice-based one which makes it a quantum-friendly solution, despite ours.

Paper structure. The structure of the rest of the paper is as follows. In Sec. 2, the preliminaries for the paper are reviewed. In Sec. 3, a definition of a CP-ABE scheme and its security is given. In Sec. 4, the proposed basic CP-ABE scheme is detailed and its security is proved. Sections 5 and 6 describe the two improved versions of the basic scheme, which are with or without the property of hidden attributes, respectively. A comparison of the proposed scheme with Boneh's scheme is brought in Sec. 7, and finally Sec. 8 concludes our work.

2. Preliminaries and Definitions

At the first we introduce the notation that we will use throughout the paper. We use λ as a security parameter. We assume that $\mathsf{negl}(\lambda)$ is the negligible function. The cardinality of set $\mathbb{A}$ is denoted by $|\mathbb{A}|$. The sets $\{1, \ldots, n\}$ and $\{0, 1, \ldots, n\}$ are denoted by $[n]$ and $[0, n]$, respectively. When we want to say x is selected uniformely random from set χ, denote by $x \leftarrow_{\$} \chi$. By $\mathbf{x}^{\mathbf{u}}$, where $\mathbf{x} = [x_0, x_1, \ldots, x_n]$ and $\mathbf{u} = [u_0, u_1, \ldots, u_n]$, we mean $\prod_{i=1}^{n} x_i^{u_i}$. Two computationally indistinguishable

distributions $\mathcal{A}$ and $\mathcal{B}$ are denoted by $\mathcal{A} \approx_c \mathcal{B}$. Also, PPT represented from "Probabilistic Polynomial Time".
Next, we provide a list of hardness assumptions.

Definition 1 (k-Multilinear map [12]). *The multilinear map is defined over k groups $\mathbb{G}_1, \mathbb{G}_2, \dots, \mathbb{G}_k$ of the same order. Assume that g_i is the generator of $\mathbb{G}_i$ for $i \in \{1, 2, \dots, k\}$. The function $e_{i,j}$ is defined as below:*

$$\begin{aligned} &e_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j};\ i, j \in [k-1];\ i + j \leq k \\ &e_{i,j}(g_i^a, g_j^b) = g_{i+j}^{ab} \end{aligned} \tag{1}$$

We can summarize the consecutive computations of several bilinear maps (1) into the following formula.

$$e(g_{i_1}^{x_1}, g_{i_2}^{x_2}, \dots, g_{i_m}^{x_m}) = g_n^{\prod_{i=1}^m x_i} \tag{2}$$

where $n = \sum_{j=1}^m i_j \leq k$. We assume that there is a polynomial-time algorithm for computing (1). The bilinear map (or pairing) is a special case of k-multilinear map for $k = 2$. throughout this paper, by $Mult_k$, we mean the following tuple.

$$Mult_k = \{\mathbb{G}_1, \dots, \mathbb{G}_k, g_1, \dots, g_k, \{e_{i,j}\}_{i,j\in[k-1]}\} \tag{3}$$

Definition 2 (k-Multilinear Diffie-Hellman assumption (k-MDH) [12]). *This assumption states that given vector $\left\{Mult_k, g^s, g^{c_1}, g^{c_2}, \dots, g^{c_k}\right\}$, where $g = g_1$, it is hard to compute $T = g_k^{s\cdot\prod_{i=1}^k c_i}$.*

Definition 3 (k-Multilinear Decisional Diffie-Hellman assumption (k-MDDH) [12]). *This assuption states that given vector $\left\{Mult_k, g^s, g^{c_1},\ g^{c_2}, \dots, g^{c_k}, g_k^z\right\}$, where $g = g_1$, it is hard to decide if $z = s \cdot \prod_{i=1}^k c_i$.*

Definition 4 ($(k-1)$-Distance Diffie-Hellman assumption ($(k-1)$-DsDH)). *This assumption states that given $\left\{Mult_k, g^x, g_k^y\right\}$, it is hard to compute $T = g_k^{x.y}$.*

Theorem 1. *The $(k-1)$-DsDH assumption is at least as hard as the k-MDH assumption.*

Proof. Given an oracle $\mathcal{O}$, which on input $\left\{Mult_k, g^x, g_k^y\right\}$ outputs $\left\{g_k^{x.y}\right\}$, we show that there exists an algorithm $\mathcal{A}$, which on input $\left\{Mult_k, g^x, g^{c_1}, \dots, g^{c_k}\right\}$ outputs $g_k^{x.\prod_{i=1}^k c_i}$. Given a vector $\left\{Mult_k, g^x, g^{c_1}, \dots, g^{c_k}\right\}$, we set $h_1 = g^x$ and $h_2 =$

$e(g^{c_1}, g^{c_2}, \dots, g^{c_k}) = g_k^{\prod_{i=1}^k c_i} = g_k^y$. We view (h_1, h_2) as an input to $\mathcal{O}$ to obtain $\mathcal{O}(h_1, h_2) = g_k^{x.y}$. It follows that $\mathcal{A}$ can compute $g_k^{x \cdot \prod_{i=1}^k c_i}$ using $\mathcal{O}$ in polynomial time with the same advantage. □

Definition 5 ((k−1)-Distance Decisional Diffie-Hellman assumption ((k−1)-DsDDH))**.** *This assumption states that given the vector* $\left\{Mult_k, g^x, g_k^y, g_k^z\right\}$*, it is hard to decide if* $z = x \cdot y$*. The advantage of algorithm* $\mathcal{A}$ *for solving the* $(k-1)$*-DsDDH problem is* $\mathbf{Adv}_{\mathcal{A},(k-1)-DsDDH}^{Distinguish} = |p - \frac{1}{2}|$*, where* p *is the success probbaility of* $\mathcal{A}$*.*

Theorem 2. *The* $(k-1)$*-DsDDH assumption is at least as hard as the* k*-MDDH assumption.*

The claim of Theorem 2 can be proved similar to the $(k-1)$-DsDH hardness proof given in the proof of Theorem 1.

3. Overview and Security Definitions

In this section, we bring the formal definition of a ciphertext-policy attribute-based encryption scheme and its security.

Definition 6. *(Arithmetic Access Function (Structure, Policy or circuit)) Suppose* q *is a large prime number. The general form of the arithmetic access function of degree at most* d *over* $\mathbb{Z}_q$ *is as follows.*

$$f(\mathbf{x}) = \sum_{\substack{\mathbf{u}_i \in [0,d]^n \\ \sum_{j\in[n]} u_{i,j} \leq d}} a_i \mathbf{x}^{\mathbf{u}_i} \tag{4}$$

where $a_i \in \mathbb{Z}_q$*,* $\mathbf{x} = [x_1, x_2, \dots, x_n]$*, and* $\mathbf{u}_i = [u_{i,1}, \dots, u_{i,2}]$*. If we define* $P_i = \{j \in [n] | u_{i,j} \neq 0\}$*, (4) can be rewritten as:*

$$f(\mathbf{x}) = \sum_{\substack{u_{i,j} \in [d] \\ \sum_{j\in P_i} u_{i,j} \leq d}} \big(a_i \prod_{j\in P_i} x_j^{u_{i,j}}\big) \tag{5}$$

We define $\mathbb{S} = \{P_i | a_i \neq 0\}$*.*

A CP-ABE scheme for arithmetic circuits realizes an access policy consistent with all or a class of the arithmetic functions defined in (5), where each x_i , $i = 1, 2, \dots, n$ corresponds to one attribute and $\mathbf{x}$ is called the attribute vector.

Definition 7. *(Ciphertext-Policy Attribute-Based Encryption scheme for arithmetic circuits): Suppose that $\mathbb{U}$ is the set of all attributes from $\mathbb{Z}_q$, where $|\mathbb{U}| = n$. And also, Σ_c is the the set of all arithmetic access functions and $\Sigma_k = 2^{\mathbb{U}}$ is the key indices over the attribute space $\mathbb{U}$ The CP-ABE scheme $\mathcal{ABE}$ for an arithmetic function* $\textsc{Af} : \Sigma_k \times \Sigma_c \to \mathbb{Z}_q$ *over message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$, is a quadruple of PPT algorithms,* $(\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$*, described in the following.*

- $(\mathsf{pp}, \mathsf{pk}, \mathsf{msk}) \leftarrow \mathcal{ABE}.\mathsf{Setup}(\lambda, k, \mathbb{U})$: *The setup algorithm generates the public parameters* pp*, the public key* pk *and the master secret key* msk *according to its inputs. The inputs are the security parameter λ, the attribute space $\mathbb{U}$, and the circuit depth k.*

- $(\mathsf{dk}_{\mathbb{B}}) \leftarrow \mathcal{ABE}.\mathsf{KGen}(\mathsf{msk}, \mathbb{B}, \mathbf{x}_{\mathbb{B}})$: *The key generation algorithm returns the decryption key* $\mathsf{dk}_{\mathbb{B}}$ *according to its inputs. The inputs of this algorithm are the master secret key* msk*, an authorized key index $\mathbb{B} \in \Sigma_k$, and the value vector* $\mathbf{x}_{\mathbb{B}} \in \mathbb{Z}_q^n$*. Note that $x_j = 0$ for all $j \notin \mathbb{B}$.*

- $(\mathtt{Ctx}_f) \leftarrow \mathcal{ABE}.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, m, f, y)$: *The Encryption algorithm outputs the ciphertext* $\mathtt{Ctx}_f \in \mathcal{C}$ *according to its inputs. Its inputs are the public parameters* pp*, public key* pk*, message $m \in \mathcal{M}$, the arithmetic access function $f \in \Sigma_c$ and a value $y \in \mathbb{Z}_q$, called the result value.*

- $\{m', \perp\} \leftarrow \mathcal{ABE}.\mathsf{Dec}(\mathsf{pp}, \mathsf{pk}, \mathtt{Ctx}_f, f, \mathsf{dk}_{\mathbb{B}}, \mathbb{B})$: *The decryption algorithm returns $m' \in \mathcal{M}$ if* $\textsc{Af}(\mathbf{x}_{\mathbb{B}}, f) = y$ *otherwise $\perp$. The inputs of this algorithm are the public parameters* pp*, the public key* pk*, the ciphertext* $\mathtt{Ctx}_f \in \mathcal{C}$ *and the corresponding access function $f \in \Sigma_c$ along with a private decryption key* $\mathsf{dk}_{\mathbb{B}}$ *for the key index $\mathbb{B} \in \Sigma_k$*

In the following, we give the definitions of the correctness of a CP-ABE scheme, and the IND-CPA security (Indistinguishability under Chosen Plaintext Attack) in the adaptive security model.

Definition 8 (Correctness)**.** *Let Ψ be a CP-ABE scheme for arithmetic functions. We say that Ψ over message space $\mathcal{M}$ and ciphertext space $\mathcal{C}$ is correct if for all $m \in \mathcal{M}$, $\mathbb{B} \in \Sigma_k$, $f \in \Sigma_c$, and $y \in Z_q$, it holds that:*

$$\Pr\left[\begin{array}{l}(\mathsf{pp}, \mathsf{pk}, \mathsf{msk}) \leftarrow \Psi.\mathsf{Setup}(\lambda, k, \mathbb{U}), \mathsf{dk}_{\mathbb{B}} \leftarrow \Psi.\mathsf{KGen}(\mathsf{msk}, \mathbb{B}, \mathbf{x}_{\mathbb{B}}), \\ \mathtt{Ctx}_f \leftarrow \Psi.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, m, f, y), \Psi.\mathsf{Dec}\left(\mathsf{pp}, \mathsf{pk}, \mathtt{Ctx}_f, \mathsf{dk}_{\mathbb{B}}, \mathbb{B}\right) = m : \\ \textsc{Af}(\mathbf{x}_{\mathbb{B}}, f) = y\end{array}\right] \approx_c 1 \ .$$

Definition 9 (Indistinguishability under Chosen Plaintext Attack (IND-CPA) in adaptive security model)**.** *Suppose that the ABE scheme Ψ is defined for the attribute space $\mathbb{U}$, message space $\mathcal{M}$ and an arrithmetic function $\textsc{Af} : \Sigma_k \times \Sigma_c \to \mathbb{Z}_q$. Suppose that the scheme Ψ is defined for the attribute space $\mathbb{U}$, message space $\mathcal{M}$ and an arrithmetic function $\textsc{Af} : \Sigma_k \times \Sigma_c \to \mathbb{Z}_q$. The adaptive security model define as below. Note that this model is described for a security parameter λ, a circuit depth k, the challenger $\mathcal{C}$.*

- ***Initialization:*** *The Challenger $\mathcal{C}$ runs $\Psi.\mathsf{Setup}(\lambda, k, \mathbb{U})$ algorithm and generates the triple of public parameters, the public key and the master secret key. Then forwards pp and pk to $\mathcal{A}$, while keeping msk secure.*

- ***First Query Phase:*** *The adversary $\mathcal{A}$ queries $\mathsf{dk}_{\mathbb{B}}$ from $\mathcal{C}$ by choosing a key index $\mathbb{B} \in \Sigma_k$ for polynomially-many requests. $\mathcal{C}$ chooses the vector $\mathbf{x}_{\mathbb{B}}$, executes algorithm $\Psi.\mathsf{KGen}(\mathsf{msk}, \mathbb{B}, \mathbf{x}_{\mathbb{B}})$, and returns $\mathsf{dk}_{\mathbb{B}}$ to $\mathcal{A}$. Also adds $\mathbb{B}$ to a list, called $\mathcal{Q}_k$. This list was initialized as an empty list.*

- ***Challenge:*** *In this phase, two messages $(m_0, m_1) \leftarrow_{\$} \mathcal{M} \times \mathcal{M}$ that have the same length, along with a challenge access function $f^* \in \Sigma_c$ are chosen by $\mathcal{A}$. Then $\mathcal{A}$ sends $\{(m_0, m_1), f^*\}$ to $\mathcal{C}$. Then, $\mathcal{C}$ flips a fair coin, produces a random bit $b \leftarrow_{\$} \{0, 1\}$, chooses $y \in \mathbb{Z}_q$ such that $\textsc{Af}(\mathbf{x}_{\mathbb{B}}, f^*) \neq y$ for all $\mathbb{B} \in \mathcal{Q}_k$, runs $\Psi.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, m_b, f^*, y)$ and sends $\mathtt{Ctx}_{f^*}$ back to $\mathcal{A}$.*

- ***Second Query Phase:*** *$\mathcal{A}$ is still allowed to repeat Query Phase, which was defined in **First Query Phase** step, after receiving the challenge ciphertext. Note that the requested keys conditioned that $\textsc{Af}(\mathbf{x}_{\mathbb{B}}, f^*) \neq y$.*

- ***Guess.*** *$\mathcal{A}$ returns a bit $b' \in \{0, 1\}$ to $\mathcal{C}$.*

Definition 10 (Indistinguishability under Chosen Plaintext Attack (IND-CPA) in adaptive security model)**.** *Suppose that the $\mathcal{A}$ ran the game defined in 9. If the advantage for all PPT adversaries $\mathcal{A}$, is negligible we say Ψ is* IND-CPA *secure. The advantage of $\mathcal{A}$ is defined as follows.*

$$\mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{A},\Psi}(1^\lambda, b) = \left| \Pr\left[b = b'\right] - \frac{1}{2} \right| \tag{6}$$

In other words, Ψ is the IND-CPA *secure if the following relationship is established.*

$$\left| \mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{A},\Psi}(1^\lambda, b = 0) - \mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{A},\Psi}(1^\lambda, b = 1) \right| \approx_c 0 \ .$$

Remark 1. *Note that, there is two main security model for ABE constructions including selective security and adaptive security. The selective security model is the weaker one. Because in this model, the adversary selects the challenge access function f^* at the beginning of the game and sends it to the challenger. Then, the challenger generates the public parameters according to the received challenge policy.*
We will prove the security of our schemes in the Adaptive security model. This feature is one of the advantages of our scheme.

4. Basic CP-ABE Scheme

This section describes a simplified and basic version of the proposed CP-ABE scheme for arithmetic access functions. The goal of these simplifications is to make it more convenient to understand the main schemes proposed in the next sections.

4.1. Features

Suppose that the circuit depth of the scheme is k. For the basic CP-ABE proposed in this section, we restrict $f(\mathbf{x})$ defined in (5) to those which $\forall i, j, u_{i,j} \in \{0, 1\}$, $d = n = k$[1], and $\forall P_i, P_j \in \mathbb{S}, i \neq j, P_i \cap P_j = \emptyset$. The proposed scheme works for any result value $y \in Z_q$. Moreover, in this scheme, the user does not know the value of his/her own attribute vector as well as the value of the result. These constraints will be relaxed in the schemes proposed in the next sections.

4.2. Specifications

The proposed CP-ABE scheme Ψ_0 is a quadruple $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ of PPT algorithms, which are described in the following.

- $\Psi_0.\mathsf{Setup}(\lambda, k, \mathbb{U})$. This algorithm takes security parameter λ, the circuit depth k, and the attribute space $\mathbb{U}$ as input. Then, it outputs the public parameters, the public key, and the master secret key. The public parameters are $\mathsf{pp} = \{Mult_k\}$ defind in (3). Then, $t_i \leftarrow_\$ Z_q, s_i \leftarrow_\$ Z_q, i \in [k]$ are choosen, and the public key pk and the master secret key msk are generated, as below.

$$\mathsf{pk} = \left\{ \{g^{t_i}, g^{\frac{1}{s_i}}, g^{\frac{t_i}{s_i}}\}_{i \in [k]}, h = g_k^{\prod_{v=1}^{k} t_v} \right\}$$

[1] Note that although the basic scheme is described for $d = n = k$, it can support functions with $d \leq k$ and $n \leq k$. For the latter case, we consider that a dummy term $\prod_{j \in [k]} x_j$ with zero coefficient is included in $f(\mathbf{x})$ descryption.

$$\mathsf{msk} = \big\{\{t_i\}_{i\in[k]}, \{s_i\}_{i\in[k]}\big\} \tag{7}$$

- $\Psi_0.\mathsf{KGen}(\mathsf{msk}, \mathbb{B}, \mathbf{x}_\mathbb{B})$. This algorithm takes the master secret key msk, key index $\mathbb{B}$, and the attribute value vector $\mathbf{x}_\mathbb{B} \in \mathbb{Z}_q^n$ as input, where $x_i = 0$ for all $i \notin \mathbb{B}$. Then, it outputs the user's secret key $\mathsf{dk}_\mathbb{B}$ as follows.

$$\mathsf{dk}_\mathbb{B} = \{\mathbb{B}, \{sk_i = s_i \cdot x_i\}_{i\in\mathbb{B}}\}^{2} \tag{8}$$

- $\Psi_0.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, m, f, y)$. This algorithm takes public key pk, arithmetic function f consistent with the specifications given in Sec. 4.1, the result value $y \in \mathbb{Z}_q$ and message m encoded to an element of G_k as inputs, then it generates $\mathtt{Ctx}_f$ as follows.

  Firstly, $r_j \leftarrow_\$ Z_q, j \in [k]$ such that $\forall P_i \in \mathbb{S}$, $\prod_{j\in P_i} r_j = R$. Note that since P_is are disjoint, such a set of $\{r_j\}_{j\in[k]}$ always exists. Then, $\{C_i\}_{i\in[k]}$ are computed as $C_i = g^{\frac{r_i t_i}{s_i}}, i \in [k]$. C_0 and check are also computed as follows.

$$\begin{aligned} C_0 &= m \cdot h^{y.R} \\ \mathsf{check} &= g_k^y \end{aligned} \tag{9}$$

  Finally, the ciphertext is returned by $\Psi_0.\mathsf{Enc}$ algorithm as below.

$$\mathtt{Ctx}_f = \big\{f, C_0, \{C_i\}_{i\in[k]}, \mathsf{check}\big\} \tag{10}$$

  The parameter check is left in the $\mathtt{Ctx}_f$ to allow the $\Psi_0.\mathsf{Dec}$ algorithm to check iff $f(\mathbf{x}_\mathbb{B}) = y$.

- $\Psi_0.\mathsf{Dec}(\mathsf{pp}, \mathsf{pk}, \mathtt{Ctx}_f, f, \mathsf{dk}_\mathbb{B}, \mathbb{B})$. This is a deterministic algorithm that takes the public paratmeters pp, public key pk, ciphertext $\mathtt{Ctx}_f$, and the users secret key $\mathsf{dk}_\mathbb{B}$ as inputs. It outputs message m only if $\mathtt{Ctx}_f$ is an encryption of m under the public key pk and $f(\mathbf{x}_\mathbb{B}) = y$ otherwise it outputs $\perp$.

  The algorithm $\Psi_0.\mathsf{Dec}$, first checks if $\mathsf{check} = g_k^{f(\mathbf{x})}$ to make sure that the input decryption key $\mathsf{dk}_\mathbb{B}$ is valid for decryption. For this purpose, it computes $g_k^{f(\mathbf{x})}$

[2] This way of defining the secret keys does not make this scheme vulnerable to the collusion attack. The reason for that will be discussed more at the end of this section

using pk and dk, as follows (for simplicity $\mathbf{x}_{\mathbb{B}}$ is denoted by $\mathbf{x}$).

$$\begin{aligned}
\mathsf{check}' &= \prod_{P_i \in \mathbb{S}} e\Big((g^{\frac{1}{s_{i_1}}})^{sk_{i_1}}, \ldots, (g^{\frac{1}{s_{i_{|P_i|}}}})^{sk_{i_{|P_i|}}}\Big)^{a_i} \\
&= \prod_{P_i \in \mathbb{S}} e\Big(g^{x_{i_1}}, \ldots, g^{x_{i_{|P_i|}}}\Big)^{a_i} \\
&= \prod_{P_i \in \mathbb{S}} g_k^{a_i \prod_{j \in P_i} x_j} \\
&= g_k^{f(\mathbf{x})} \qquad (11)
\end{aligned}$$

where $P_i = \{i_j\}_{j \in [|P_i|]}$. If $\mathsf{check}' = \mathsf{check}$, this algorith decrypts the ciphertext as follows, otherwise it returns $\perp$. For decryption, the algorithm first computes $I_{P_i}, P_i \in \mathbb{S}$ as follows.

$$I_{P_i} = e(C_{i_1}, C_{i_2}, \ldots, C_{i_{|P_i|}}, g^{t_{j_1}}, g^{t_{j_2}}, \ldots, g^{t_{j_{(k-|P_i|)}}}) \qquad (12)$$

where $\{j_1, \ldots, j_{k-|P_i|}\} = [k] \setminus P_i$. Then, it computes mask, and decrypts the ciphertext $\mathtt{Ctx}_f$ into message m' as follows.

$$\begin{aligned}
\mathsf{mask} &= \prod_{P_i \in \mathbb{S}} (I_{P_i})^{a_i \prod_{j \in P_i} sk_j} \\
m' &= \frac{C_0}{\mathsf{mask}} \qquad (13)
\end{aligned}$$

Correctness. The correctness of equation (13) is as follows. We first simplify (12) using to the following equality.

$$\begin{aligned}
I_{P_i} &= g_k^{\prod_{j \in P_i} (\frac{r_j . t_j}{s_j}) \cdot \prod_{v \notin P_i} t_v} \\
&= g_k^{\frac{\prod_{j \in P_i} (r_j)}{\prod_{j \in P_i} (s_j)} \cdot \prod_{v=1}^{k} t_v} = h^{\frac{\prod_{j \in P_i} (r_j)}{\prod_{j \in P_i} (s_j)}} \\
&= h^{\frac{R}{\prod_{j \in P_i} (s_j)}} \qquad (14)
\end{aligned}$$

So, mask would be equal to

$$\begin{aligned}
\mathsf{mask} &= \prod_{P_i\in\mathbb{S}} (I_{P_i})^{a_i \prod_{j\in P_i} sk_j} \\
&= \prod_{P_i\in\mathbb{S}} \left(h^{\frac{R}{\prod_{j\in P_i}(s_j)}}\right)^{a_i \prod_{j\in P_i} s_j x_j} \\
&= \prod_{P_i\in\mathbb{S}} h^{R\cdot a_i(\prod_{j\in P_i}(x_j))} \\
&= h^{R\left(\sum_{P_i\in\mathbb{S}}\left(a_i\cdot \prod_{j\in P_i} x_j\right)\right)} = h^{f(\mathbf{x}).R} \qquad (15)
\end{aligned}$$

Finally, equations (15) along with (9) yeilds (13).

Example 1. *Assume that* $\mathbb{S} = \{P_1, P_2\}$ *where* $P_1 = \{1, 3\}$ *and* $P_2 = \{2\}$*. Here,* $k = n = 3$ *and* $f(\mathbf{x}) = a_1x_1x_3 + a_2x_2$*.* mask *is simplified as follows.*

$$\begin{aligned}
\mathsf{mask} &= \prod_{i=1}^{2} (I_{P_i})^{a_i \prod_{j\in P_i} sk_j} \\
&= (I_{P_1})^{a_1 \prod_{j\in\{1,3\}} sk_j} \cdot (I_{P_2})^{a_2 \prod_{j\in\{2\}} sk_j} \\
&= (I_{P_1})^{a_1(s_1x_1.s_3x_3)} \cdot (I_{P_2})^{a_2(s_2x_2)} \\
&= h^{R\cdot a_1x_1x_3} \cdot h^{R\cdot a_2x_2} \\
&= h^{R(a_1x_1x_3+a_2x_2)} = h^{f(\boldsymbol{x}).R}
\end{aligned}$$

Note that in this scheme the non-eligible users can not effectively collude to decrypt an impermissible ciphertext. Since the value of attributes, as well as the result, is unknown to the users, they can not realize which combination of secret keys can lead to successful collusion.

4.3. Security

In this section, we prove that the basic scheme proposed in Sec. 4.2 is adaptively IND-CPA secure under the $(k-1)$-DsDDH assumption.

Theorem 3. *The proposed basic CP-ABE scheme described in Sec. 4.2, for arithmetic functions with the characteristics given in Sec. 4.1 achieves IND-CPA in adaptive security model under* $(k-1)$*-DsDDH assumption.*

Proof. Suppose that there exists a polynomial-time attacker $\mathcal{A}$ for the proposed basic CP-ABE scheme with non-negligible advantage in the IND-CPA security game (Def. 9 and 10). Under this assumption, there exist a polynomial-time algorithm $\mathcal{C}$ that uses the adversary $\mathcal{A}$ as a black-box and solves an instance of the $(k-1)$-DsDDH problem with non-negligible advantage.

We suppose that the oracle $\mathcal{D}$ generates the $(k-1)$-DsDDH parameters as $\{Mult_k, g^x, g_k^y, g_k^z\}$. $\mathcal{D}$ flips a fair coin μ and sets $z = x \cdot y$ if $\mu = 0$ else $z \leftarrow_{\$} \mathbb{Z}_q$. The challenger $\mathcal{C}$ gets the $(k-1)$-DsDDH parameters and, by a blackbox access to $\mathcal{A}$, it aims to distinguish if $z = x \cdot y$ or it is a random value and return his guess μ', with non-negligible advantage. The security game for proof of the basic scheme is as follows.

- **Initialization:** The challenger $\mathcal{C}$ chooses $t_i \leftarrow_{\$} \mathbb{Z}_q, i \in [k-1]$, and $s_i \leftarrow_{\$} \mathbb{Z}_q, i \in [k]$. Then, it sets $g^{t_k} = g^{x \cdot \prod_{i=1}^{k-1} t_i^{-1}}$, and simulates the public parameters and public key for the attacker $\mathcal{A}$ as follows.

$$\begin{aligned} \mathsf{pp} &= \{Mult_k\} \\ \mathsf{pk} &= \Big\{\{g^{t_i}\}_{i\in[k-1]}, g^{t_k} = g^{x\cdot\prod_{i=1}^{k-1} t_i^{-1}}, \{g^{\frac{1}{s_i}}\}_{i\in[k]}, \\ &\qquad \{g^{\frac{t_i}{s_i}}\}_{i\in[k-1]}, g^{\frac{t_k}{s_k}} = g^{\frac{x\cdot\prod_{i=1}^{k-1} t_i^{-1}}{s_k}}, h = e_{1,k-1}(g^x, g_{k-1}) = g_k^x\Big\} \end{aligned}$$

- **First Query Phase.** After receiving pp and pk, $\mathcal{A}$ requests $\mathcal{C}$ for secret keys associated to its chosen key index $\mathbb{B} \in \Sigma_k$. The challenger $\mathcal{C}$ chooses an $\mathbf{x}_{\mathbb{B}} \leftarrow_{\$} \mathbb{Z}_q^k$ where $x_i = 0$ for $i \notin \mathbb{B}$, as the attribute vector and generates the secret key according to (8) by simulating the $\Psi_0.\mathsf{KGen}(\mathsf{msk}, \mathbb{B}, \mathbf{x}_{\mathbb{B}})$ algorithm. Then, it sends it to $\mathcal{A}$, upon each secret key requested by $\mathcal{A}$. $\mathcal{C}$ adds the recieved key index $\mathbb{B}$ to list $\mathcal{Q}_k$.

- **Challenge.** $\mathcal{A}$ chooses two same length messages $(m_0, m_1) \leftarrow_{\$} \mathcal{M} \times \mathcal{M}$ and the challenge access function f^* and sends$\{(m_0, m_1), f^*\}$ to $\mathcal{C}$. $\mathcal{C}$ flips a faircoin, generating the random bit b, chooses $y \in \mathbb{Z}_q$ such that $\textsc{Af}(\mathbf{x}_{\mathbb{B}}, f^*) \neq y$ for all $\mathbb{B} \in \mathcal{Q}_k$. Then, $\mathcal{C}$ runs algorithm $\Psi_0.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, m, f^*, y)$ to simulate the ciphertext $\mathtt{Ctx}_{f^*}$ of m_b for $\mathcal{A}$ as below.

$$\mathtt{Ctx}_{f^*} = \big\{f^*, C_0, \{C_i\}_{i\in[k]}, \mathsf{check}\big\}$$

where $C_0 = m_b.(g_k^z)^R$, $C_i = g^{\frac{r_i t_i}{s_i}}$, for $i \in [k-1]$, $C_k = g^{x r_k s_k^{-1} \cdot \prod_{i=1}^{k-1} t_i^{-1}}$, and $\mathsf{check} = g_k^y$. The challenger $\mathcal{C}$ then sends $\mathtt{Ctx}_{f^*}$ to $\mathcal{A}$.

- **Second Query Phase.** Having received $\mathtt{Ctx}_{f^*}$, $\mathcal{A}$ can adaptively request more secret keys associated with more key indices $\mathbb{B}$. $\mathcal{C}$ chooses $\mathbf{x}_{\mathbb{B}}$ such that $\textsc{Af}(\mathbf{x}_{\mathbb{B}}, f^*) \neq y$, generates the requested keys, and sends them to $\mathcal{A}$.
- **Guess.** The attacker $\mathcal{A}$ sends the guessed bit b' of b to the $\mathcal{C}$. If $b' = b$, $\mathcal{C}$ will output $\mu' = 0$ indicating that $z = xy$ in the given $(k-1)$-DsDDH instance, otherwise it outputs $\mu' = 1$ indicating that the given $(k-1)$-DsDDH instance was a random tuple.

The advantage of $\mathcal{C}$ for solving the $(k-1)-$DsDDH problem is computed as follows. In the case that $\mu = 1$, $\mathcal{A}$ gains no information about b. Therefore, we have $Pr[b' \neq b|\mu = 1] = \frac{1}{2}$. Since $\mathcal{C}$ guesses $\mu' = 1$ when $b \neq b'$, we have $Pr[\mu' = \mu|\mu = 1] = \frac{1}{2}$.

If $\mu = 0$ then $\mathcal{A}$ sees an encryption of m_b. Suppose that the advantage of $\mathcal{A}$ in this situation is the non-negligible value ϵ. Therefore, we have $Pr[b = b'|\mu = 0] = \frac{1}{2} + \epsilon$. Since $\mathcal{C}$ guesses $\mu = 0$ when $b = b'$, we have $Pr[\mu' = \mu|\mu = 0] = \frac{1}{2} + \epsilon$. The overall advantage of $\mathcal{C}$ in the $(k-1)$-DsDDH game is:

$$\begin{aligned}\mathbf{Adv}^{Distinguish}_{\mathcal{C},(k-1)-\text{DsDDH}} &= \frac{1}{2}Pr[\mu' = \mu|\mu = 0] + \frac{1}{2}Pr[\mu' = \mu|\mu = 1] - \frac{1}{2} \\ &= \frac{1}{2} \cdot (\frac{1}{2} + \epsilon) + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{\epsilon}{2} \qquad (16)\end{aligned}$$

In (16), the probability of resolving $(k-1)-$DsDDH problem is non-negligibly greater than $\frac{1}{2}$. So, it is concluded that attacker $\mathcal{A}$ does not exist, since $(k-1)-$DsDDH problem is assumed to be hard. □

5. Hidden-Result and Hidden-Attributes CP-ABE Scheme

In this section, we propose an improved version of the basic CP-ABE scheme for arithmetic circuits, proposed in Sec. 4, where all the limitations of the basic scheme over the access function are relaxed. The result value and the value of the attribute vector in this scheme both are hidden to the user.

5.1. Features

The arithmetic function that can be realized as the access structure in this scheme is in the general form of (5) with no constraint on n, P_is, and $u_{i,j}$. It means that the constraint of $\forall i \neq j, P_i \cap P_j = \emptyset$ is relaxed, and for circuit depth k, n can be greater than k, and d is at most equal to k. So, $u_{i,j} \in [0, k]$, conditioned that $\sum_{j \in P_i} u_{i,j} \leq k$.

5.2. Specifications

The quadruple of algorithms ($\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$) of this version of the proposed CP-ABE scheme is similar to the basic scheme's, introduced in Sec. 4.2, except for the following modifications in. In this section, Ψ_1 refers to the proposed hidden-result and hidden-attribute CP-ABE scheme.

- $\Psi_1.\mathsf{Setup}(\lambda, k, \mathbb{U})$. The public parameters, the public key and the master secret key are generated as below.

$$\begin{aligned} \mathsf{pp} &= \{Mult_k\} \\ \mathsf{pk} &= \left\{ \left\{ g^{t_i}, g^{\frac{1}{s_j}}, g^{\frac{t_i}{s_j}} \right\}_{i\in[k], j\in[n]}, h = g_k^{\prod_{v=1}^{k} t_v} \right\} \\ \mathsf{msk} &= \left\{ \{t_i\}_{i\in[k]} \{s_i\}_{i\in[n]} \right\} \end{aligned} \tag{17}$$

  where, $t_i \leftarrow_{\$} Z_q, i \in [k]$ and $s_j \leftarrow_{\$} Z_q, j \in [n]$.

- $\Psi_1.\mathsf{KGen}(\mathsf{msk}, \mathbb{B}, \mathbf{x}_{\mathbb{B}})$. This algorithm is the similar to $\Psi_0.\mathsf{KGen}$ algorithm. The only difference is in the size of the user secret key vector, which can reach up to n:

$$\mathsf{dk}_{\mathbb{B}} = \{\mathbb{B}, \{sk_j = s_j \cdot x_j\}_{j\in\mathbb{B}}\}$$

- $\Psi_1.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, m, f, y)$: This algorithm takes public parameters pp and public key pk, the arithmetic function f, the result value $y \in \mathbb{Z}_q$ and message m which is encoded to an element of G_k, as inputs then outputs the ciphertext $\mathtt{Ctx}_f$.

  Firstly, the random numbers $r_j^{(i)} \in Z_q$, where $j \in P_i$ and $P_i \in \mathbb{S}$ are selected in a way that for all i it holds $\prod_{j\in P_i} r_j^{(i)} = R$. The ciphertext is then computed according to the following equation.

$$\mathtt{Ctx}_f = \{f,\ C_0, \{\mathbf{C}_{P_i}\}_{P_i\in\mathbb{S}}, \mathsf{check}\} \tag{18}$$

  where $C_0 = m \cdot h^{y.R}$ and $\mathsf{check} = g_k^y$.

$$\mathbf{C}_{P_i} = [C_1^{(i)}, C_2^{(i)}, \ldots, C_{|P_i|}^{(i)}], \quad \forall P_i \in \mathbb{S} \tag{19}$$

  and $C_j^{(i)} = g^{\frac{r_j^{(i)} t_j}{s_j}}$, for $j \in P_i$ and $P_i \in \mathbb{S}$.

Algorithm 1: Computing I_{P_i}

```
Input: pp, pk, $P_i$, $\{u_j\}_{j \in P_i}$ and $\mathbf{C}_{P_i}$
Output: $I_{P_i}$
$B \leftarrow e(C_1^{(i)}, C_2^{(i)}, \ldots, C_{|P_i|}^{(i)})$;
$T \leftarrow \{1, \ldots, k\} \setminus P_i$;
for $j \in P_i$ do
    for $k \in [u_j - 1]$ do
        select $i' \in T$;
        $B \leftarrow e(B, g^{\frac{t_{i'}}{s_j}})$;
        $T \leftarrow T \setminus \{i'\}$;
while $T \neq \emptyset$ do
    select $i' \in T$;
    $B \leftarrow e(B, g^{t_{i'}})$;
    $T \leftarrow T \setminus \{i'\}$;
return $B$;
```

- Ψ_1.Dec(pp, pk, Ctx$_f$, f, dk$_{\mathbb{B}}$, $\mathbb{B}$): The only change in the Ψ_1.Dec algorithm is in I_{P_i} formula, $P_i \in \mathbb{S}$. In this version, it is more convenient to use an algorithmic presentation to explain how I_{P_i} is computed, rather than a closed-form formula. So, Algorithm 1 is run to get I_{P_i}. According to this algorithm, I_{P_i} is returned as follows.

$$I_{P_i} = g_k^{\frac{R}{\prod_{j \in P_i} s_j^{u_j}} \prod_{v=1}^{k} t_v} \tag{20}$$

The rest of the Ψ_1.Dec algorithm is exactly similar to Ψ_0.Dec in the basic scheme. We bring an example here to show how Algorithm 1 works.

Example 2. *Suppose that $k = 7$ and the i^{th} monomial of $f(x)$ is $x_1^3 x_2^2 x_4$. So,*

$P_i = \{1, 2, 4\}$ and $u_1 = 3, u_2 = 2$ and $u_4 = 1$. Algorithm 1 computes I_{P_i} as follows.

$$\begin{aligned} I_{P_i} &= e(C_1^{(i)}, C_2^{(i)}, C_4^{(i)}, g^{\frac{t_3}{s_1}}, g^{\frac{t_5}{s_1}}, g^{\frac{t_6}{s_2}}, g^{t_7}) \\ &= (g^{\frac{r_1^{(i)} t_1}{s_1}}, g^{\frac{r_2^{(i)} t_2}{s_2}}, g^{\frac{r_4^{(i)} t_4}{s_4}}, g^{\frac{t_3}{s_1}}, g^{\frac{t_5}{s_1}}, g^{\frac{t_6}{s_2}}, g^{t_7}) \\ &= g_7^{\frac{r_1^{(i)} r_2^{(i)} r_4^{(i)}}{s_1^3 s_2^2 s_4} \prod_{v=1}^{k} t_v} \\ &= g_7^{\frac{R}{s_1^3 s_2^2 s_4} \prod_{v=1}^{k} t_v} \end{aligned} \tag{21}$$

5.3. Security

The IND-CPA security of the proposed scheme, in the adaptive security model, is reduced to the $(k-1)$-DsDDH assumption.

Theorem 4. *The improved hidden-result hidden-attribute CP-ABE scheme described in Sec. 5.1 for arithmetic functions with the characteristics given in Sec. 5.1 achieves IND-CPA in the adaptive security model, under $(k-1)$-DsDDH assumption.*

Proof. The security proof of this scheme is completely similar to the security proof of the basic scheme given in Sec. 4.3. □

6. Hidden-Result Disclosed-Attributes CP-ABE Scheme

In the two previous schemes, the attribute vector is hidden from its owner. Depending on the application, such a property may be desired or not. In this section, we present a variant of the proposed scheme in which the values of the attributes are known to the attribute-owner.

6.1. Features

Like the scheme proposed in Sec. 5, The access functions supported by this scheme are in the most general form of (5). Contrary to the two previous schemes, in this scheme, the attribute vector is included in $\mathsf{dk}_{\mathbb{B}}$, i.e., it is known to its owner. On the other hand, the result value, y, is hidden before the decryption, but if $\textsc{Af}(\mathbf{x}_{\mathbb{B}}, f) = y$, the value of y will be disclosed in Dec algorithm. In other words, the eligible user who can successfully decrypt the ciphertext can obtain the result value after decryption.

For a circuit depth of k, this scheme requires a $2k$-multilinear map. This increases the size of public parameters and secret keys as well as the computational complexity of the decryption algorithm but does not affect on the public key and master secret key sizes.

6.2. Specifications

In this section, we mention only those parts of algorithms (Setup, KeyGen, Enc, Dec) that have changed comparing to the proposed scheme in Sec. 5.2.

- $\Psi_2.\mathsf{Setup}(\lambda, n, \mathbb{U})$. The public parameters, the public key and the master secret key are returned by this algorithm, as below.

$$\begin{aligned} \mathsf{pp} &= \{mult_{2k}\} \\ \mathsf{pk} &= \left\{ \left\{ g^{t_i}, g^{\frac{1}{s_j}}, g^{\frac{t_i}{s_j}} \right\}_{i\in[k], j\in[n]}, h = g_{2k}^{\prod_{v=1}^{k} t_v} \right\} \\ \mathsf{msk} &= \left\{ \{t_i\}_{i\in[k]} \{s_i\}_{i\in[n]} \right\} \end{aligned} \tag{22}$$

where, $t_i \leftarrow_\$ Z_q, i \in [k]$ and $s_j \leftarrow_\$ Z_q, j \in [n]$. Note that in this scheme $h = g_{2k}^{\prod_{v=1}^{k} t_v}$.

- $\Psi_2.\mathsf{KGen}(\mathsf{msk}, \mathbb{B}, \mathbf{x}_\mathbb{B})$: This algorithm first selects $\alpha \leftarrow_\$ \mathbb{Z}_q$, then returns the secret key, $\mathsf{dk}_\mathbb{B}$, as below.

$$\mathsf{dk}_\mathbb{B} = \{\mathbb{B},\ \mathbf{x}_\mathbb{B},\ sk_{1,j},\ sk_{2,j}\}_{j\in\mathbb{B}} \tag{23}$$

where $sk_{1,j} = s_j \cdot x_j \cdot (x_j)^\alpha,\ sk_{2,j} = g^{x_j^{-\alpha}}$.

- $\Psi_2.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, f, m, y)$: This algorithm is the same as $\Psi_1.\mathsf{Enc}(\mathsf{pp},\ \mathsf{pk}, f, m, y)$ algorithm. The only change in this algorithm is as follows.

$$\begin{aligned} C_0 &= m \cdot h^{y \cdot R} \\ \mathsf{check} &= g_{2k}^{y} \end{aligned} \tag{24}$$

- $\Psi_2.\mathsf{Dec}(\mathsf{pp}, \mathsf{pk}, \mathtt{Ctx}_f, f, \mathsf{dk}_\mathbb{B}, \mathbb{B})$. The computation of check' is much more simple than the previous schemes. The value of $g_{2k}^{f(\mathbf{x}_\mathbb{B})}$ can be easily computed using the attribute vector $\mathbf{x}_\mathbb{B}$ included in $\mathsf{dk}_\mathbb{B}$. Then, it is compared to the received check value. If $\mathsf{check}' \neq \mathsf{check}$, the algorithm returns $\perp$ and y remains unknown, otherwise the it is revealed that $y = f(\mathbf{x}_\mathbb{B})$ and of the decryption is proceeds as follows.

The value of $I_{P_i}, P_i \in \mathbb{S}$ is computed according to Algorithm 1. Then, given $\mathsf{dk}_{\mathbb{B}}$ and $\mathtt{Ctx}_f$, $J_{P_i}, P_i \in \mathbb{S}$ is computed as follows.

$$\begin{aligned} J_{P_i} &= e(\underbrace{sk_{2,j_1}, \ldots, sk_{2,j_1}}_{u_{j_1} \text{ times}}, \ldots, \underbrace{sk_{2,j_{|P_i|}}, \ldots, sk_{2,j_{|P_i|}}}_{u_{j_{|P_i|}} \text{ times}}, \underbrace{g, \ldots, g}_{k-k_i \text{ times}}) \\ &= g_k^{\prod_{j\in P_i} x_j^{-u_j\alpha}} \end{aligned} \tag{25}$$

where $P_i = \{j_1, \ldots, j_{|P_i|}\}$, and $k_i = \sum_{j\in|P_i|} u_j$. Finally, mask is computed as follows.

$$\begin{aligned} \mathsf{mask} &= \prod_{P_i\in\mathbb{S}} e_{k,k}(I_{P_i}^{a_i \prod_{j\in P_i} sk_{1,j}^{u_j}}, J_{P_i}) \\ &= \prod_{P_i\in\mathbb{S}} e_{k,k}(g_k^{R\cdot a_i \prod_{j\in P_i} x_j^{u_j} (x_j)^{u_j\alpha} \prod_{v=1}^{k} t_v}, g_k^{\prod_{j\in P_i} x_j^{-u_j\alpha}}) \\ &= \prod_{P_i\in\mathbb{S}} h^{R\cdot a_i \prod_{j\in P_i} x_j^{u_j}} = h^{R\cdot \sum_{P_i\in\mathbb{S}} a_i \prod_{j\in P_i} x_j^{u_j}} \\ &= h^{R\cdot f(\mathbf{x})} \end{aligned} \tag{26}$$

The rest of the $\Psi_2.\mathsf{Dec}$ algorithm is similar to $\Psi_1.\mathsf{Dec}$ given in 5.2.

6.3. Security

The security proof of this scheme is mostly similar to the security proof of the basic scheme brought in Sec. 4.3, but with some modifications. However, we bring the complete security proof in this section.

Theorem 5. *The improved hidden-result disclosed-attribute CP-ABE scheme described in Sec. 6.2 for arithmetic functions with the characteristics given in 6.1 achieves IND-CPA in the adaptive security model under the* $(2k-1)$*-DsDDH assumption.*

Proof. We suppose that the oracle $\mathcal{D}$ generates the $(2k-1)$-DsDDH parameters as $\{Mult_{2k}, g^x, g_{2k}^y, g_{2k}^z\}$. $\mathcal{D}$ flips fair coin μ and sets $z = x \cdot y$ if $\mu = 0$ else $z \leftarrow_{\$} \mathbb{Z}_q$. The challenger $\mathcal{C}$ gets the $(2k-1)$-DsDDH parameters and, by running the IND-CPA game, it aims to distinguish if $z = x \cdot y$ or it is a random value and return his guess μ', with non-negligible advantage. The security game for proof of the second improved scheme is as follows.

- **Initialization.** The challenger $\mathcal{C}$ chooses $t_i \leftarrow_{\$} \mathbb{Z}_q, i \in [k-1]$, and $s_i \leftarrow_{\$} \mathbb{Z}_q, i \in [n]$. Then, it sets $g^{t_k} = g^{x \cdot \prod_{i=1}^{k-1} t_i^{-1}}$ and $h = e_{1,2k-1}(g^x, g_{2k-1}) = g_{2k}^x$, and simulates the public parameters and public key for the attacker $\mathcal{A}$ as follows.

$$\begin{aligned} \mathsf{pp} &= \{Mult_{2k}\} \\ \mathsf{pk} &= \{\{g^{t_i}\}_{i\in[k-1]}, g^{t_k} = g^{x \cdot \prod_{i=1}^{k-1} t_i^{-1}}, \{g^{\frac{1}{s_i}}\}_{i\in[n]}, \\ &\quad \{g^{\frac{t_i}{s_j}}\}_{\substack{i\in[k-1],\\ j\in[n]}}, \{g^{\frac{t_k}{s_j}} = g^{\frac{x \cdot \prod_{i=1}^{k-1} t_i^{-1}}{s_j}}\}_{j\in[n]}, h = g_{2k}^x\} \end{aligned}$$

- **First Query Phase.** After receivingthe public parameters and public key, $\mathcal{A}$ requests $\mathcal{C}$ for secret keys associated to its chosen attribute vector $\mathbf{x}_{\mathbb{B}} \in \mathbb{Z}_q^n$. The challenger $\mathcal{C}$ generates $\mathsf{dk}_{\mathbb{B}}$ by simulating the $\Psi_2.\mathsf{KGen}(\mathsf{msk}, \mathbb{B}, \mathbf{x}_{\mathbb{B}})$ algorithm (8). Then, it sends it to $\mathcal{A}$. $\mathcal{C}$ addes the recieved key index $\mathbb{B}$ to list $\mathcal{Q}_k$. This step can be reapeted adaptively to simulate the collusion of users.

- **Challenge.** $\mathcal{A}$ chooses two same length messages $(m_0, m_1) \leftarrow_{\$} \mathcal{M} \times \mathcal{M}$ and the challenge access function f^* and sends $\{(m_0, m_1), f^*\}$ to $\mathcal{C}$. $\mathcal{C}$ flips a faircoin, generating the random bit b, chooses $y \in \mathbb{Z}_q$ such that $\textsc{Af}(\mathbf{x}_{\mathbb{B}}, f^*) \neq y$ for all $\mathbb{B} \in \mathcal{Q}_k$. Then, $\mathcal{C}$ runs algorithm $\Psi_2.\mathsf{Enc}(\mathsf{pp}, \mathsf{pk}, m, f^*, y)$ to simulate the ciphertext $\mathtt{Ctx}_{f^*}$ of m_b for $\mathcal{A}$ as below.

$$\mathtt{Ctx}_{f^*} = \{f^*, C_0, \{\mathbf{C}_{P_i}\}_{P_i \in \mathbb{S}}, \mathsf{check}\}$$

where $C_0 = m_b.(g_{2k}^z)^R$, $\mathsf{check} = g_{2k}^y$, and $\{\mathbf{C}_{P_i}\}_{P_i \in \mathbb{S}}$ are computed according to (19). The challenger $\mathcal{C}$ then sends $\mathtt{Ctx}_{f^*}$ to $\mathcal{A}$.

- **Second Query Phase.** Having received $\mathtt{Ctx}_{f^*}$, $\mathcal{A}$ can adaptively request more secret keys associated with new attribute vectors $\mathbf{x}_{\mathbb{B}}$. Although $\mathcal{A}$ chooses $\mathbf{x}_{\mathbb{B}}$, the probability of $\textsc{Af}(\mathbf{x}_{\mathbb{B}}, f^*) = y$ is negligible. $\mathcal{C}$ generates the requested keys, and sends them to $\mathcal{A}$.

- **Guess.** The attacker $\mathcal{A}$ sends the guessed bit b' of b to the $\mathcal{C}$. If $b' = b$, $\mathcal{C}$ will output $\mu' = 0$ indicating that $z = x \cdot y$ in the given $(2k-1)$-DsDDH instance, otherwise it outputs $\mu' = 1$ indicating it was a random tuple.

The overall advantage of $\mathcal{C}$ in the $(2k-1)$-DsDDH game is:

$$\begin{aligned}\mathbf{Adv}^{Distinguish}_{\mathcal{C},(2k-1)-\text{DsDDH}} &= \frac{1}{2}Pr[\mu' = \mu|\mu = 0] + \frac{1}{2}Pr[\mu' = \mu|\mu = 1] - \frac{1}{2} \\ &= \frac{1}{2}\cdot(\frac{1}{2}+\epsilon) + \frac{1}{2}\cdot\frac{1}{2} - \frac{1}{2} = \frac{\epsilon}{2}\end{aligned} \tag{27}$$

In (27), the probability of resolving $(2k-1)-$DsDDH problem is non-negligibly greater than $\frac{1}{2}$. So, it is concluded that attacker $\mathcal{A}$ does not exist, since $(2k-1)-$DsDDH problem is assumed to be hard. □

7. Comparison with Boneh's scheme

The only ABE scheme for arithmetic functions so far is the scheme of Boneh et al. [7]. Although this work is a lattice-based scheme with the benefit of being a post-quantum CP-ABE scheme, the proposed schemes in this paper have some other advantages over that, which are listed in the following.

1. Despite Boneh's scheme which has selective security, the proposed scheme is adaptively secure.

2. The proposed scheme is CP-ABE which is more flexible than KP-ABE.

3. The both scenarios of hidden- and disclosed- attribute vector can be supported by the proposed scheme. However, in Boneh's scheme, the attribute vector can not be kept hidden.

4. In Boneh's scheme, the values of attributes must be in $[-p, p]$, where p is less than the group order q, for **Multiply** gates. But, the proposed schemes do not put any constraint on the attribute values.

5. The arithmetic function supported by the proposed scheme is more general than Boneh's scheme. Our scheme supports the exponentiation gate. However, it seems that this feature can be added to Boneh's scheme, as well.

6. Since Boneh's scheme is a lattice-based scheme, the computational complexity, and the key size are larger than our scheme's.

7. The result parameter in the proposed schemes is an arbitrary-chosen value. But, Boneh's scheme just works for $y = 0$ while not supporting a non-zero a_0 in the access function. However, it seems to be modifiable to work for an arbitrary result value.

8. Conclusion

We proposed three variants of a CP-ABE scheme for arithmetic circuit access functions. The proposed scheme relies on multilinear maps. We defined the new concept of hidden-result ABE which refers to an ABE scheme for arithmetic functions with unknown result values.

We first proposed a basic CP-ABE scheme for arithmetic functions, in which the attribute vector and the result value are hidden to the users. For a circuit depth k, this scheme requires a k-multilinear map and supports a number of $n = k$ attributes. Then, an improved hidden-result and hidden-attribute CP-ABE scheme was proposed which works for any number of $n \geq k$ attributes, conditioned that the degree of the function is at most k. Finally, we proposed an improved hidden-result and disclosed-attribute CP-ABE scheme for the access functions like the previous scheme, which based on a $2k$-multilinear map.

We proved that these schemes are adaptively secure under a newly defined hardness assumption, called the k-Distance Decisional Diffie-Hellman problem, which is at least as hard as the well-known k-multilinear decisional Diffie-Hellman problem. Finally, we compared our schemes with Boneh et al.'s scheme and described the advantages of ours.

Acknowledgement

This work was supported by the Iranian National Science Foundation (INSF) under Contract no. 99021948.

References

[1] Shweta Agrawal, Xavier Boyen, Vinod Vaikuntanathan, Panagiotis Voulgaris, and Hoeteck Wee. Fuzzy identity based encryption from lattices. IACR Cryptol. ePrint Arch., 2011:414, 2011.

[2] Amel Arfaoui, Soumaya Cherkaoui, Ali Kribeche, and Sidi Mohammed Senouci. Context-aware adaptive remote access for iot applications. IEEE Internet of Things Journal, 7(1):786-799, 2019.

[3] Nuttapong Attrapadung and Hideki Imai. Dual-policy attribute based encryption. In International Conference on Applied Cryptography and Network Security, pages 168-185. Springer, 2009.

[4] Nuttapong Attrapadung, Benoît Libert, and Elie De Panafieu. Expressive key-policy attribute-based encryption with constant-size cipher-texts. In International workshop on public key cryptography, pages 90-108. Springer, 2011.

[5] Sana Belguith, Nesrine Kaaniche, Maryline Laurent, Abderrazak Jemai, and Rabah Attia. Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot. Computer Networks, 133:141-156, 2018.

[6] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In 2007 IEEE symposium on security and privacy (SP'07), pages 321-334. IEEE, 2007.

[7] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 533-556. Springer, 2014.

[8] Xavier Boyen. Attribute-based functional encryption on lattices. In Theory of Cryptography Conference, pages 122-142. Springer, 2013.

[9] Melissa Chase. Multi-authority attribute based encryption. In Theory of cryptography conference, pages 515-534. Springer, 2007.

[10] Hui Cui, Robert H Deng, and Guilin Wang. An attribute-based framework for secure communications in vehicular ad hoc networks. IEEE/ACM Transactions on Networking, 27(2):721-733, 2019.

[11] Hua Deng, Zheng Qin, Qianhong Wu, Zhenyu Guan, and Yunya Zhou. Flexible attribute-based proxy re-encryption for efficient data sharing. Information Sciences, 511:94-113, 2020.

[12] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai, and BrentWaters. Attribute-based encryption for circuits from multilinear maps. In Annual Cryptology Conference, pages 479-499. Springer, 2013.

[13] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute- based encryption for circuits. Journal of the ACM (JACM), 62(6):1-33, 2015.

[14] Vipul Goyal, Omkant Pandey, Amit Sahai, and BrentWaters. Attribute- based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM conference on Computer and communications security, pages 89-98, 2006.

[15] Matthew Green, Susan Hohenberger, Brent Waters, et al. Outsourcing the decryption of abe ciphertexts. In USENIX security symposium, volume 2011, 2011.

[16] Junbeom Hur and Dong Kun Noh. Attribute-based access control with efficient revocation in data outsourcing systems. IEEE Transactions on Parallel and Distributed Systems, 22(7):1214-1221, 2010.

[17] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. Journal of cryptology, 26(2):191-224, 2013.

[18] Venkata Koppula and Brent Waters. Realizing chosen ciphertext security generically in attribute-based encryption and predicate encryption. In Annual International Cryptology Conference, pages 671-700. Springer, 2019.

[19] Jiguo Li, Xiaonan Lin, Yichen Zhang, and Jinguang Han. KSF-OAB: Outsourced attribute-based encryption with keyword search function for cloud storage. IEEE Transactions on Services Computing, 10(5):715- 725, 2016.

[20] Jiguo Li, Qihong Yu, and Yichen Zhang. Hierarchical attribute based encryption with continuous leakage-resilience. Information Sciences, 484:113-134, 2019.

[21] Rafail Ostrovsky, Amit Sahai, and Brent Waters. Attribute-based en- cryption with non-monotonic access structures. In Proceedings of the 14th ACM conference on Computer and communications security, pages 195-203, 2007.

[22] Xuanmei Qin, Yongfeng Huang, Zhen Yang, and Xing Li. LBAC: A lightweight blockchain-based access control scheme for the internet of things. Information Sciences, 554:222-235, 2021.

[23] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In Annual international conference on the theory and applications of cryp- tographic techniques, pages 457-473. Springer, 2005.

[24] Haijiang Wang, Jianting Ning, Xinyi Huang, Guiyi Wei, Geong Sen Poh, and Ximeng Liu. Secure fine-grained encrypted keyword search for e-healthcare cloud. IEEE, 2019.

[25] Peng Wang, Tao Xiang, Xiaoguo Li, and Hong Xiang. Access control encryption without sanitizers for internet of energy. Information Sciences, 546:924-942, 2021.

[26] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In International Workshop on Public Key Cryptography, pages 53-70. Springer, 2011.

[27] Jianghong Wei, Xiaofeng Chen, Xinyi Huang, Xuexian Hu, and Willy Susilo. Rs-habe: Revocable-storage and hierarchical attribute-based access scheme for secure sharing of e-health records in public cloud. IEEE Transactions on Dependable and Secure Computing, 2019.

[28] Hu Xiong, Yanan Zhao, Li Peng, Hao Zhang, and Kuo-Hui Yeh. Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing. Future Generation Computer Systems, 97:453-461, 2019.

[29] Shengmin Xu, Jiaming Yuan, Guowen Xu, Yingjiu Li, Ximeng Liu, Yinghui Zhang, and Zuobin Ying. Efficient ciphertext-policy attribute- based encryption with blackbox traceability. Information Sciences, 538:19-38, 2020.

[30] Jiang Zhang and Zhenfeng Zhang. A ciphertext policy attribute-based encryption scheme without pairings. In International Conference on Information Security and Cryptology, pages 324-340. Springer, 2011.

[31] Xiubin Zou. A hierarchical attribute-based encryption scheme. Wuhan University Journal of Natural Sciences, 18(3):259-264, 2013.

[32] Ge, Chunpeng, Willy Susilo, Joonsang Baek, Zhe Liu, Jinyue Xia, and Liming Fang. "Revocable attribute-based encryption with data integrity in clouds." IEEE Transactions on Dependable and Secure Computing (2021).