Journal of Computers Vol. 31 No. 1, 2020, pp. 134-147
doi:10.3966/199115992020023101011

Multi-Authority Attribute-based Encryption Resilient against Auxiliary-Input Leakage

Hai-Ying Ma[1], Zhan-Jun Wang[2*], Jin-Hua Wang[2], Zhi-Jin Guan[1]

[1] School of Computer Science and Technology, Nantong University, Nantong 226019, Jiangsu, China
{mhy8855, guan.zj}@ntu.edu.cn

[2] School of Science, Nantong University, Nantong 226019, Jiangsu, China
{wzj8855, jhwang}@ntu.edu.cn

Received 29 May 2018; Revised 19 September 2018; Accepted 18 November 2018

Abstract. The existing multi-authority Attribute-Based Encryption (ABE) schemes cannot tolerate the secret key leakages under a variety of side-channel attacks. We integrate the auxiliary-input leakage model with dual system encryption technique, and reasonably design the generations of secret key and ciphertext, then propose a multi-authority attribute-based encryption resilient against auxiliary-input leakage. Based on the modified Goldreich-Levin theorem and the subgroup decision problem assumptions, we prove our scheme to be fully secure even if the adversary can obtain the leakages on attribute-based private keys with the auxiliary input functions. Compared with the relative leakage-resilient ABE schemes, our scheme not only achieves numeric unbounded leakages on the attribute-based private key of users, but also support the multi-authority application scenarios. Therefore, our scheme can effectively resist a larger possible class of potential attackers in the decentralizing multi-authority ABE systems.

Keywords: auxiliary-input leakage model, dual system encryption, multi-authority attribute-based encryption, subgroup decision problem assumptions

1 Introduction

Attribute-based encryption (ABE) can achieve fine-grained access control over shared data, which has drawn considerable attention in a large scale distributed system such as public cloud computing and Internet of Things. In an ABE system, one central authority issues the private key for each user according to their attributes or credentials, and the data owners can specify an access policy as a boolean formula on a set of attributes, then encrypt the data with this policy. Users can decrypt a ciphertext only if the users' attributes satisfy the access policy of ciphertext. Sahai and Waters [1] first introduces the notion of ABE, then ABE is expanded in two forms: Ciphertext-Policy ABE (CP-ABE) and Key-Policy ABE (KP-ABE). In CP-ABE [2], user's private key is related to his own attributes, and each ciphertext is embedded into an access policy specified by the data owner. Whereas, KP-ABE [3] embed an access policy into users' private key, and ciphertext is labeled with the set of attributes. In most of ABE schemes, all attributes of users must belong to the same trusted domain, and are supervised by only one central authority. However, in many global scale systems, the attributes of users can come from different trust domains and organizations, then one single authority cannot verify the attributes across different domains and organizations.

To overcome the above problem, Chase [4] first uses a trusted Central Authority (CA) and global identifiers to construct a multi-authority ABE (MA-ABE). In the MA-ABE, the CA can decrypt every ciphertext, which can compromise the security and privacy of users. To enhance user's privacy in MA-ABE, Chase and Chow [5] utilize a distributed pseudo random function to remove the central authority, and propose a MA-ABE with user privacy. Gorasia et al. [6] provide a MA-ABE which allows fast

* Corresponding Author

decryption. Their MA-ABE schemes [4-6] are only proven to be selectively secure. To improve the security of MA-ABE, Lewko and Waters [7] propose a fully secure multi-authority CP-ABE without any central authority. In their scheme, the user's global identifier is used to tie private key components together, and any party can act as an authority that supervises a set of attributes and may issue attribute private keys to different users without any global coordination. To protect the privacy of user's communications, Wang et al. [8] propose a distributed MA-ABE scheme to efficiently achieve privacy-preserving in mobile social networks. So far, all of the existing MA-ABE schemes are proven secure under the assumption that any attacker cannot obtain any leakage information about secret key and other internal state. However, in practical applications of MA-ABE systems, the side-channel attackers can learn partial information about secret keys by observing the physical features of many cryptographic operations such as timing consumption, cold boot attacks, etc. [9-11]. Therefore, the existing MA-ABE schemes which are proven "secure" may be vulnerable in practice.

Our motivation is to further enhance the security of MA-ABE such that MA-ABE are proven secure against the largest possible class of potential attackers. Leakage-resilience cryptography may provide formal security guarantees in the presence of the leakage of the secret key, and allows the attacker to specify a computable leakage function $F(\cdot)$ and obtain the output of $F(\cdot)$ applied to the secret key. Obviously, the necessary restriction of leakage function $F(\cdot)$ is to prevent the attacker from learning the entire secret key. One leakage model [12] allows the attacker to learn a subset of the bits representing the secret key or internal state. Another auxiliary-input leakage model [13] specifies a class of one-way functions, and allows the attacker to use the one-way functions to reveal the whole secret key *sk*, but the attacker still cannot recover *sk* from $F(sk)$. Therefore, the auxiliary-input leakage model is more desirable, since it specifies the least necessary restriction on the leakage functions. From the existing literature, both of the leakage models have been employed in many one-authority ABE systems [14-19], and have not been employed in multi-authority ABE systems.

1.1 Our Contributions

In this work, we propose the first multi-authority CP-ABE that remains fully secure even if the attacker obtains the leakage information of the secret key with any auxiliary-input function. We borrows the construction technique [13] to extend Lewko and Waters's decentralizing ABE [7]. To resist the secret key leakage with auxiliary input functions, we split the master secret key into m pieces, where $m = (3\log p_2)^{1/\varepsilon}$, p_2 is a large prime of λ-bit number, $0 < \varepsilon < 1$, and reasonably design the generations of private key and ciphertext. To prove the adaptive security of our scheme, we transfer normal secret keys and ciphertexts into semi-functional ones. In the hybrid argument games, the ciphertext is first turned into semi-functional, then the private keys are turned into semi-functional one by one, and we prove that these changes are indistinguishable. There exists an important challenge problem in the indistinguishability of games: the simulator cannot confirm the nature of ciphertext by testing a semi-functional private key. To overcome this challenge, we set the simulator to construct a nominal semi-functional ciphertext: it is distributed like a semi-functional ciphertext in the attacker's view, if the simulator use semi-functional keys to decrypt it, decryption often succeeds. Then the attacker will not distinguish the semi-functional ciphertext from the nominally semi-functional one.

To combine the dual system with the modified Goldreich-Levin theorem, we must restrict the blind factor of the semi-functional private key to be a number belonging to $[0, \lambda]$, which is different from the blind factor of LW's semi-functional key [14]. Without this restriction, the simulator's running time is $O(2^{\lambda})$, which is undesirable. As the auxiliary input function must be ascertained before the challenge ciphertext is disclosed, the attacker cannot distinguish the leakage on a nominally semi-function ciphertext from that on a common semi-function ciphertext. This allows us to achieve leakage resilience in the multi-authority ABE system. Therefore, our scheme not only combines the benefits of auxiliary-input leakage resilience and dual system encryption, but also achieves numeric unbounded leakage from the attribute-based private key of users in the multi-authority CP-ABE circumstance.

1.2 Related Work

To meet the practical application requirements of multiple authorities, Chase [4] first introduced the notion of multiple authority and proposed a MA-ABE scheme. Since a central authority that is utilized in

the scheme can collect each user's attributes and decrypt each ciphertext, the security and privacy of users can be compromised. Chase and Chow [5] utilize the pseudo random functions to remove the central authority. Li et al. [20] presented a multi-authority CP-ABE which can trace traitors who release their decryption keys to others. The constructions of MA-ABE are only proven selectively secure [4-6, 20]. In a selective model of security, the attacker must immediately submit its target rather than adaptively choosing it during the security game. Lewko and Waters [7] propose a fully secure multi-authority CP-ABE without central authority. Ma et al. [21] constructed a fully secure MA-ABE scheme which can adaptively trace pirates. The existing MA-ABE schemes are proven secure under the assumption of secret key's absolute security [4-8, 20, 21]. However, the attack can use a variety of side-channel attacks [9-11] to obtain some information of secret key or internal state. Therefore, the existing MA-ABE systems cannot resist the side-channel attacks and may be vulnerable in practice.

Exposure-resilient cryptography may guarantee the security of cryptographic constructions even if the attacker obtains the leakage of secret key or internal state. A variety of leakage models are proposed in previous works [11-13, 22, 25-27]. Micali and Reyzin [22] introduced the model of only computation leakage, which assumes the leakage happens when the device performs a cryptographic computation, but any part information of memory not related to the computation cannot be leaked. Akavia et al. [11] introduced the bounded retrieval model, which allows attackers to learn leakage information on memory contents without assumption that only computation can leak information. This model assumes the total amount of leakage during the lifetime of system is dramatically less than the bit-size of secret key, and it is employed in many cryptographic constructions [12, 23-24]. But this model cannot allow a user to update his secret key over the lifetime of system. Recently, the continual leakage model was proposed to allow attackers to learn leakage between the updates of secret key [25-26], and assumes the amount of leakage between successive updates is bounded by a small fraction of secret key size. But this model cannot allow leakage during the update process. Lewko et al. [14] combined the continual leakage with the technology of dual system encryption to improve the leakage tolerance of ABE constructions. Two works also constructed several ABE schemes resilient against continual leakage [17, 19].

The auxiliary input leakage model was first proposed by Dodis et al. [13] to further loose the restriction of leakage functions. This model specifies a kind of computationally irreversible functions to simulate a large class of leakage process. Although the irreversible functions can information-theoretically reveal the whole secret key, any polynomial time attackers cannot recover the secret key by using the functions. The auxiliary input model has been employed in the ABE constructions [15]. Yuen et al. [27] combined the continual memory leakage with auxiliary inputs model to propose a continual auxiliary leakage model, and used this model to construct leakage-resilient ABE scheme. Ma et al. [16] combined the continual auxiliary leakage model with the technology of dual system encryption, and constructed an ABE resilient against continuous auxiliary-inputs leakage. So far, the existing leakage-resilient ABE systems only supports single authority application scenarios, does not support multi-authority ones.

1.3 Organization

We give the necessary definitions and the complexity assumptions in section 2. The definition and security model of our multi-authority CP-ABE resilient against auxiliary-input leakage is defined in section 3. We propose a multi-authority CP-ABE scheme resilient against auxiliary-input leakage in section 4. We prove our scheme by using the modified Goldreich-Levin theorem and the dual system encryption technology, and give performance analysis in section 5. Finally, we conclude this work in section 6.

2 Preliminaries

2.1 Notation

Let N and m be a positive integer, G be a cyclic group of order N, and denote by $x \in Z_N$ the fact that x is picked uniformly at random from the field Z_N, and by $x, y, z \in Z_N$ that all x, y, z are chosen uniformly at

random and independently from the field Z_N. Let PPT denote a Probabilistic Polynomial-Time algorithm, $|x|$ denote the number of bits of terms x.

Let angle brackets $\langle \cdot, \cdot, \cdots, \cdot \rangle$ denote vectors. The dot product of vectors is denoted by $\cdot$ and component-wise multiplication is denoted by $*$. The exponentiation operator for vectors is defined as follows: for $\forall\ \boldsymbol{v} = <v_1, v_2, \cdots, v_m> \in G^m$, $u \in G$, $a \in Z_N$, $\boldsymbol{b} = <b_1, b_2, \cdots, b_m> \in Z_N^m$, we define: $u^{\boldsymbol{b}} := \left\langle u^{b_1}, u^{b_2}, \cdots, u^{b_m} \right\rangle$, $\boldsymbol{v}^a := \left\langle v_1^a, v_2^a, \cdots, v_m^a \right\rangle$. Then we define a bilinear pairing operation of vectors in G_m : for $\forall\ \boldsymbol{v} = <v_1, v_2, \cdots, v_m> \in G^m$, and $\boldsymbol{w} = <w_1, w_2, \cdots, w_m> \in G^m$, their pairing is $e(\boldsymbol{v}, \boldsymbol{w}) = e(v_1, w_1)\ e(v_2, w_2) \cdots e(v_m, w_m) = \prod_{i=1}^{m} e(v_i, w_i)$.

2.2 Composite Order Bilinear Groups and Computational Assumptions

We define a group generator $\mathcal{G}(\cdot)$ which takes in a security parameter λ and outputs a description of a composite order bilinear group $(p_1, p_2, p_3, G, G_T, e(\cdot, \cdot))$, where p_1, p_2, p_3 are three distinct primes, G and G_T are cyclic groups of order $N = p_1p_2p_3$, e: $G \times G \rightarrow G_T$ is a map so that: (1) Bilinear: $\forall\ g, h \in G$, $a, b \in Z_N$, $e(g^a, h^b) = e(g, h)^{ab}$. (2) Non-degenerate: $\exists\ g \in G$ such that $e(g, g)$ has order N in G_T. (3) Computable: The bilinear map and group operation are efficiently computable in polynomial time.

Let G_1, G_2, G_3, $G_{1,2}$, and $G_{1,3}$ denote subgroups of order p_1, p_2, p_3, p_1p_2, p_1p_3 in G respectively. According to this orthogonal property [7] of subgroups G_1, G_2 and G_3, when $h_i \in G_i$ and $h_j \in G_j$, for $i \neq j$, $e(h_i, h_j)$ is the identity element in G_T, where $i = 1, 2, 3$, and $j = 1, 2, 3$. We will utilize this orthogonal property to construct our scheme and prove its security. We list the four subgroup decision assumptions [7] and the Goldreich-Levin Theorem on any field $GF(q)$ [13], which are used to prove the security of our scheme. Let $g \leftarrow G_1$ denote the variable g is selected randomly from the subgroup G_1, $Pr(\cdot)$ is the probability function, and λ is a security parameter in our system.

Assumption 1. Given a group generator $\mathcal{G}(\cdot)$, which outputs the following random distributions $\mathbb{G} = (N = p_1p_2p_3, G, G_T, e(\cdot, \cdot)) \leftarrow \mathcal{G}(\lambda)$, and randomly pick $g_1 \in G_1$, $T_1 \in G$, $T_2 \in G_1$. Let $E = (\mathbb{G}, g_1)$, The advantage of any PPT algorithm $\mathcal{A}$ in solving Assumption 1 is defined as: $Adv1_{\mathcal{A}}(\lambda) = |Pr(\mathcal{A}(E, T_1) = 1) - Pr(\mathcal{A}(E, T_2) = 1)|$.

Assumption 2. Given the following random distributions: $\mathbb{G} = (N = p_1p_2p_3, G, G_T, e(\cdot, \cdot)) \leftarrow \mathcal{G}(\lambda)$, and randomly pick $g_1, X_1 \in G_1, X_2 \in G_2, X_3 \in G_3, T_1 \in G_1, T_2 \in G_{1,2}$. Let $E = (\mathbb{G}, g_1, g_3, X_1X_2)$. The advantage of any PPT algorithm $\mathcal{A}$ in solving Assumption 2 is defined as: $Adv2_{\mathcal{A}}(\lambda) = |Pr(\mathcal{A}(E, T_1) = 1) - Pr(\mathcal{A}(E, T_2) = 1)|$.

Assumption 3. Given the following random distributions: $\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \leftarrow \mathcal{G}(\lambda)$, and randomly pick $g_1, X_1 \in G_1, Y_2 \in G_2, X_3, Y_3 \in G_3, T_1 \in G_{1,2}, T_2 \in G_{1,3}$. Let $E = (\mathbb{G}, g_1, X_1X_3, Y_2Y_3)$. The advantage of any PPT algorithm $\mathcal{A}$ in solving Assumption 3 is defined as: $Adv3_{\mathcal{A}}(\lambda) = |Pr(\mathcal{A}(E, T_1) = 1) - Pr(\mathcal{A}(E, T_2) = 1)|$.

Assumption 4. Given the following random distributions: $\mathbb{G} = (N = p_1p_2p_3, G, G_T, e) \leftarrow \mathcal{G}$, and randomly pick $a, b, c, d \in Z_N$, $b_1 \in G_1, b_2 \in G_2, b_3 \in G_3, T_2 \in G_T$. Let $T_1 = e(b_1, b_1)^{abc}$, $E = (\mathbb{G}, g_1, g_2, g_3, b_1^a, b_1^b b_3^b, b_1^c, b_1^{ac} b_3^d)$. We define the advantage of an algorithm $\mathcal{A}$ in breaking Assumption 4 to be $Adv4_{\mathcal{A}}(\lambda) = |Pr(\mathcal{A}(E, T_1) = 1) - Pr(\mathcal{A}(E, T_2) = 1)|$.

Definition 1. We say that $\mathcal{G}(\cdot)$ satisfies Assumption 1, 2, 3, 4 if and only if $Adv1_{\mathcal{A}}(\lambda)$, $Adv2_{\mathcal{A}}(\lambda)$, $Adv3_{\mathcal{A}}(\lambda)$ and $Adv4_{\mathcal{A}}(\lambda)$ are four negligible functions of λ for any PPT attacker $\mathcal{A}$.

Theorem 1 (Goldreich-Levin Theorem on any Field *GF*(*q*) [13]**).** Suppose that q is a large prime and B is any subset of $GF(q)$. Let a function $f: B^n \to \{0, 1\}^*$, $\boldsymbol{s} \leftarrow B^n$, $\xi \leftarrow f(\boldsymbol{s})$, $\boldsymbol{r} \leftarrow GF(q)^n$. If there is a distinguisher $\mathcal{B}$ which computes in time t so that $|\Pr[\mathcal{B}(\xi, \boldsymbol{r}, (\boldsymbol{r} \cdot \boldsymbol{s})) = 1] - \Pr[\zeta \leftarrow GF(q): \mathcal{B}(\xi, \boldsymbol{r}, \zeta) = 1]| = \varepsilon$, then there exists an inventor $\mathcal{A}$ which computes in time $t' = t \cdot \text{poly}(n, |B|, 1/\varepsilon)$ so that $\Pr[\boldsymbol{s} \leftarrow B^n, \xi \leftarrow f(\boldsymbol{s}): \mathcal{A}(\xi) = \boldsymbol{s}] \geq \varepsilon^3/(512 \cdot n \cdot q^2)$.

2.3 Access Structures and Linear Secret-Sharing Schemes (LSSS)

Definition 2 (Access structures [7]). Let $\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}$ be a set of attributes. A collection $\mathbb{A} \subseteq 2^{\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}}$ is monotone if $\forall$ B, C and $B \in \mathbb{A}$, $B \subseteq C$ then $C \in \mathbb{A}$. An access structure is a collection $\mathbb{A}$ of non-empty subsets of $\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}$, *i.e.* $\mathbb{A} \subseteq 2^{\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}} \setminus \{\varnothing\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

Definition 3 (LSSS [7]). Let $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_n\}$ and ρ be a function: $\{1, 2, \cdots, l\} \to \mathcal{P}$. A secret-sharing scheme Ω over a set $\mathcal{P}$ of attributes is called linear over Z_p if (1) The shares for each attribute consist of a vector over Z_p; (2) Let a matrix $\boldsymbol{A}$ be the share-generation matrix for Ω. The matrix $\boldsymbol{A}$ has n rows and l columns. For each $i = 1, 2, \cdots, n$, the i^{th} row of $\boldsymbol{A}$ is labeled by an attribute $\rho(i)$. If we set the column vector $\boldsymbol{v} = (s, r_2, \cdots, r_l)^{\mathrm{T}}$, where $s \in Z_p$ is the secret to be shared and $r_2, \cdots, r_l \in Z_p$ are randomly picked, then $\boldsymbol{A} \cdot \boldsymbol{v}$ is the vector of n shares of the secret s according to Π. The share $(\boldsymbol{Av})_i$ belongs to the attribute $\rho(i)$.

The linear reconstruction property of LSSS: Let the secret-sharing scheme Ω be an LSSS for the access structure $\mathbb{A}$ and $\omega \in \mathbb{A}$ be any authorized set, and $I \subseteq \{1, 2, \cdots, n\}$ is defined as $I = \{i: \rho(i) \in \omega\}$. If $\{s_i\}$ are valid shares of any secret s according to secret-sharing scheme Ω, then there exist constants $\{c_i \in Z_p\}_{i \in I}$ that satisfy $\Sigma_{i \in I} c_i s_i = s$. Furthermore, it is shown that these constants c_i can be found in polynomial time in the size of the share-generating matrix $\boldsymbol{A}$. In our scheme, we define the access structure $\mathbb{A}(\boldsymbol{A}, \rho)$ for a secret-sharing scheme Ω that has a share-generating matrix $\boldsymbol{A}$ and function ρ.

3 Multi-Authority CP-ABE Resilient against Auxiliary-Input Leakage

A multi-authority CP-ABE resilient against auxiliary-input leakage consists of five algorithms as follows:
System Setup(λ) $\to$ *SP*. This algorithm inputs the security parameter λ and outputs the system public parameters *SP*.
Authority KeyGen(*SP*) $\to$ (*APK*, *ASK*). Each authority inputs the system public parameters *SP* and outputs its public key *APK* and private key *ASK*.
Enc(M, $\mathbb{A}(\boldsymbol{A}, \rho)$, {*APK*}, *SP*) $\to$ *CT*. This algorithm takes in a message M, an access structure $\mathbb{A}(\boldsymbol{A}, \rho)$, the set {*APK*} of public keys for relevant authorities, and the system parameters *SP*. It outputs a ciphertext *CT*.
User KeyGen(*SP*, *UID*, *i*, *ASK*) $\to$ $D_{i,UID}$. The authority supervising an attribute i utilizes the system parameters *SP*, user's identity *UID*, the attribute i and its private key *ASK* to generate the user's private key $D_{i,\text{UID}}$ for this attribute-identity pair (i, *UID*).
Dec(*CT*, $\{D_{i,UID}\}$, *SP*) $\to M/\perp$. This algorithm inputs the ciphertext *CT*, a set $\{D_{i,UID}\}$ of private keys with the same identity *UID*, and the system parameters *SP*. It outputs a message M or a failure notation $\perp$.

We define the security model of our scheme by the attack game between a challenger $\mathcal{C}$ and an attacker $\mathcal{A}$. Let $\mathcal{F}$ denote a set of PPT functions and S denote the set of all authorities. Each attribute can only be managed by one authority. The attacker is allowed to statically corrupt the authority, then adaptively make key queries and leakage of private keys. Additionally, the attacker can choose the public keys of the corrupted authorities for itself.

Setup. The challenger $\mathcal{C}$ runs the system setup algorithm. The attacker $\mathcal{A}$ designates a set B of corrupt authorities. For honest authorities in the set S - B, $\mathcal{C}$ runs the authority key generation algorithm to obtain their public keys APK and private keys ASK, and sends APK to $\mathcal{A}$.

Query Phase 1. $\mathcal{A}$ can query the user key-generation oracles to the challenger $\mathcal{C}$.

OUKeyGen(i, UID): $\mathcal{A}$ submits pairs (i, UID) to $\mathcal{C}$. $\mathcal{C}$ runs User KeyGen(SP, UID, i, ASK) $\rightarrow D_{i,UID}$ and sends $D_{i, UID}$ to $\mathcal{A}$.

Challenge 1. $\mathcal{A}$ submits a challenge access structure $\mathbb{A}(\boldsymbol{A}^*, \rho)$ to $\mathcal{C}$. $\mathcal{C}$ picks a set ω^* of attributes such that ω^* can satisfy the structure $\mathbb{A}(\boldsymbol{A}^*, \rho)$. Then for each attribute $i \in \omega^*$ belonging to a good authority, $\mathcal{C}$ runs the key generation algorithm of corresponding authority to obtain $D_{i, UID}$ for the user UID. $\mathcal{C}$ first checks the list $L_{(\boldsymbol{A}^*, \rho)}$. If there is no such tuple ($D_{i, UID}$, UID, i, j), where $j \geq 1$, then it puts the tuple ($D_{i, UID}$, UID, i, 1) in the list $L_{(\boldsymbol{A}^*, \rho)}$. Otherwise, the number j is set to (j+1). Especially, a user of pair (i, UID) may make another key query after it gets the first copy.

Query Phase 2. $\mathcal{A}$ can query the leakage oracles as follows:

OLeak(f, UID): Given $f \in \mathcal{F}$, this leakage oracle returns $f(L_{(\boldsymbol{A}^*, \rho)}, \{APK\}, UID)$ to $\mathcal{A}$.

Challenge 2. $\mathcal{A}$ submits two messages M_0 and M_1 to $\mathcal{C}$. $\mathcal{C}$ picks a random bit $\beta \in \{0, 1\}$ and sends a ciphertext of M_β under the access structure $\mathbb{A}(\boldsymbol{A}^*, \rho)$ to $\mathcal{A}$.

Query Phase 3. $\mathcal{A}$ can query the user key generation oracles the same as Query Phase 1, and it must ensure that all user keys of pairs (i, UID) generated in both Query Phase 1 and 3 together with the attribute set controlled by bad authorities can not satisfy the challenge the access structure $\mathbb{A}(\boldsymbol{A}^*, \rho)$ in challenge 1.

Guess. $\mathcal{A}$ outputs a guess β' for β. $\mathcal{A}$ wins if $\beta' = \beta$.

The $\mathcal{A}$'s advantage in the above game is defined as $\mathrm{Adv}_{\mathcal{A}} = \Pr[\beta' = \beta] - 1/2$.

Now we define the function families $\mathcal{F}$. Let $\mathcal{S}^*$ denote the set of all possible keys that satisfies the access structure $\mathbb{A}(\boldsymbol{A}^*, \rho)$. Let $\mathcal{S}$ denote the set of keys created in both Query Phase 1 and 3 of the security game such that $\mathcal{S}^* \cap \mathcal{S} = \varnothing$. We set $\mathcal{F}_{\{APK\}\text{-ow}}(f(k))$ to denote the class of all PPT functions F: $\{0, 1\}^* \rightarrow \{0, 1\}^*$, such that given SP, $\{APK\}$, $\mathbb{A}(\boldsymbol{A}^*, \rho)$, $\mathcal{S}$ and $F(L_{(\boldsymbol{A}^*, \rho)}, \{APK\}, UID)$ for (SP, $\{APK\}$, $D_{i,UID}$), $\mathcal{S}$, $L_{(\boldsymbol{A}^*, \rho)}$ that is randomly generated. No PPT algorithm $\mathcal{A}$ can find a valid secrect key $\{D_{i,UID}\}$ for $\mathbb{A}(\boldsymbol{A}^*, \rho)$ with probability greater than $F(k)$, where $F(k) \geq 2^{-k}$ is a difficult parameter. Let CPA denote the Chosen-Plaintext Attack, AI denote Auxiliary-Input.

Definition 5. A multi-authority attribute-based encryption resilient against auxiliary-input leakage is called to be $F(k)$-AI-CPA secure if there exists no PPT attack $\mathcal{A}$ that can win the above game against the function family $\mathcal{F}_{\{APK\}\text{-ow}}(F(k))$ with more than non-negligible advantage.

4 Multi-authority CP-ABE Resilient against Auxiliary-Input Leakage

System Setup(λ) $\rightarrow$ SP**:** In this system setup algorithm, let $0 < \varepsilon < 1$, $m = (3\log p_2)^{1/\varepsilon}$, $N = p_1 p_2 p_3$ be the order of bilinear groups G and G_T, G_1 be the subgroup of order p_1. For $j = 1, 2, \cdots, m$, this algorithm chooses a generator $g_i \in G_1$, and a hash function $H_j : \{0, 1\}^* \rightarrow G$ mapping users' identities UID to elements of G, and outputs the system public parameters $SP = \{N, g_1, g_2, \cdots, g_m, H_1, H_2, \cdots, H_m\}$.

Authority KeyGen(SP) → (APK, ASK): For each attribute i belonging to the authority, the authority chooses two random vectors $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{im}) \in Z_N^m$ and $\boldsymbol{y}_i = (y_{i1}, y_{i2}, \cdots, y_{im}) \in Z_N^m$, and publishes its public key $APK = \{\prod_{j=1}^{m} e(g_j, g_j)^{\alpha_{ij}}, \prod_{j=1}^{m} g_j^{y_{ij}}, \forall i\}$, and keeps its private key $ASK = \{\boldsymbol{\alpha}_i, \boldsymbol{y}_i, \forall i\}$.

Enc(SP, $\{APK\}$, M, $\mathbb{A}(\boldsymbol{A}, \rho)$) → CT. This algorithm takes in a message M, an $n \times l$ access matrix $\boldsymbol{A}$ with ρ mapping its rows to attributes, the set $\{APK\}$ of public keys for relevant authorities, and the system parameters SP. It chooses a random element $s \in Z_N$ and a random vector $\boldsymbol{v} \in Z_N^l$ with s as its first entry. Let λ_x denote $A_x \cdot \boldsymbol{v}$, where A_x is row x of the matrix $\boldsymbol{A}$. It also chooses a random vector $\boldsymbol{w} \in Z_N^l$ with 0 as its first entry, and computes $\omega_x = A_x \cdot \boldsymbol{w}$. For each row A_x of $\boldsymbol{A}$ and $j = 1, 2, \cdots, m$, it chooses a random element $r_{x,j} \in Z_N$ and computes ciphertext CT as follows:

$$C_0 = M \cdot \prod_{j=1}^{m} e(g_j, g_j)^s, C_{1,x,j} = e(g_j, g_j)^{\lambda_x} \cdot \left(\prod_{j=1}^{m} e(g_j, g_j)^{\alpha_{\rho(x)j}}\right)^{r_{x,j}}, \tag{1}$$

$$C_{2,x,j} = g_j^{r_{x,j}}, C_{3,x,j} = \left(\prod_{j=1}^{m} g_j^{y_{\rho(x),j}}\right)^{r_{x,j}} g_j^{\omega_x}. \tag{2}$$

User KeyGen(SP, UID, i, ASK) → $D_{i,UID}$. To create a key of UID for attribute i belonging to an authority, the authority computes the user's private key:

$$D_{i,UID} = (g_1^{\alpha_{i1}} H_1(UID)^{y_{i1}}, g_2^{\alpha_{i2}} H_2(UID)^{y_{i2}}, \cdots, g_m^{\alpha_{im}} H_1(UID)^{y_{im}}). \tag{3}$$

Dec(CT, $\{\boldsymbol{D}_{i,UID}\}$) →$M$. Assuming the ciphertext is encrypted under an access matrix $\mathbb{A}(\boldsymbol{A}, \rho)$, if the attributes of the user's private keys satisfy the access matrix $\boldsymbol{A}$, he first computes $H_j(UID)$, and performs as follows:

$$C_{1,x,j} \cdot \frac{e(C_{3,x,j}, H_j(UID))}{\prod_{j=1}^{m} e(C_{2,x,j}, D_{\rho(x),UID,j})} = e(g_j, g_j)^{\lambda_x} \cdot e(H_j(UID), g_j)^{\omega_x}. \tag{4}$$

This algorithm then chooses constants $c_x \in Z_N$ such that $\sum_x c_x A_x = (1, 0, \cdots, 0)$ and computes:

$$C_4 = \prod_{j=1}^{m} \prod_{x} (e(g_j, g_j)^{\lambda_x} \cdot e(H_j(UID), g_j)^{\omega_x})^{c_x} = \prod_{j=1}^{m} e(g_j, g_j)^s. \tag{5}$$

Then the message can be obtained as: $M = C_0 / C_4$.

5 Security Proof and Performance Comparison

5.1 Security Proof

To prove the security of our scheme, we need transfer normal secret keys and ciphertexts into semi-functional secret keys and ciphertexts, which are not used in the real system. When the attributes of the private key satisfy the policy of ciphertext, normal keys can always decrypt both forms of ciphertext, while semi-functional keys only decrypt normal ciphertexts successfully. In the true game, all keys and the ciphertext are normal. In the hybrid argument games, the ciphertext is first turned into semi-functional, then the private keys are turned into semi-functional one by one. We will prove that these games are indistinguishable. Semi-functional ciphertexts will contain terms of subgroups G_2 and G_3. Semi-functional keys are divided into two types: Semi-functional keys of Type 1 have terms in G_2, while semi-functional keys of Type 2 have terms in G_3. To precisely describe semi-functional ciphertexts and keys, we fix random values $z_{i1}, \cdots, z_{im}, t_{i1}, \cdots, t_{im} \in Z_N$ for each attribute i, and these values will not vary for

different users in their semi-functional ciphertexts and keys. Then we let $f_1, f_2, \cdots, f_m, f$ denote generators of G_2, $h_1, h_2, \cdots, h_m, h$ denote generators of G_3.

To construct a semi-functional ciphertext, we first perform the encryption algorithm to get a normal ciphertext, C_0^*, $C_{1,x,j}^*$, $C_{2,x,j}^*$, $C_{3,x,j}^*$, $\forall x, j$. For $j = 1, 2, \cdots, m$, we select random vectors $\boldsymbol{u}_{2,j}, \boldsymbol{u}_{3,j} \in Z_N^l$, and set $\delta_{x,j}=A_x \cdot \boldsymbol{u}_{2,j}$, $\sigma_{x,j}=A_x \cdot \boldsymbol{u}_{3,j}$ for each row A_x of the access matrix $\boldsymbol{A}$. Let B denote the subset of rows of $\boldsymbol{A}$ whose corresponding attributes belong to corrupted authorities, and B' is the subset of rows of $\boldsymbol{A}$ whose corresponding attributes belong to honest authorities. For each row $A_x \in B'$, we choose random exponents $\gamma_{x,j}, \psi_{x,j} \in Z_N$. The semi-functional ciphertext is formed as:

$$C_0 = C_0^*,\ C_{1,x,j} = C_{1,x,j}^*,\ C_{2,x,j} = C_{2,x}^* \cdot f_j^{\gamma_{x,j}} \cdot h_j^{\psi_{x,j}}, \tag{6}$$

$$C_{3,x,j} = C_{3,x,j}^* \cdot \left(\prod_{j=1}^{m} f_j^{z_{\rho(x),j}}\right)^{\gamma_{x,j}} \cdot f^{\delta_{x,j}} \cdot \left(\prod_{j=1}^{m} h_j^{t_{\rho(x),j}}\right)^{\psi_{x,j}} h^{\sigma_{x,j}} \quad \forall x, \text{ s.t. } \boldsymbol{A}_x \in B', \tag{7}$$

$$C_0 = C_0^*,\ C_{1,x,j} = C_{1,x,j}^*,\ C_{2,x,j} = C_{2,x,j}^*,\ C_{3,x,j} = C_{3,x,j}^* \cdot f^{\delta_{x,j}} \cdot h^{\sigma_{x,j}} \quad \forall x, \text{ s.t. } \boldsymbol{A}_x \in B. \tag{8}$$

Semi-functional keys. For an identity *UID* and attributes i belonging to honest authorities B', we define two types of semi-functional keys. To construct a semi-functional key for *UID*, set $H^*(UID)$ be a random element of G_1, and select randomly $c_1, c_2, \cdots, c_m \in [0, \lambda]^m$. We define $H_j(UID) = H_j^*(UID) \cdot f^{c_j}$ in a semi-functional key of Type 1, and create $D_{i,UID}$ by first creating a normal key $D^*_{i,UID}$ and setting:

$$D_{i,UID} = (g_1^{\alpha_{i1}} H_1(UID)^{y_{i1}} f^{c_1 \cdot z_{i1}}, g_2^{\alpha_{i2}} H_2(UID)^{y_{i2}} f^{c_2 \cdot z_{i2}}, \cdots, g_m^{\alpha_{im}} H_m(UID)^{y_{im}} f^{c_m \cdot z_{im}}). \tag{9}$$

We define $H_j(UID) = H_j^*(UID) \cdot h^{c_j}$ in a semi-functional key of Type 2, and create $D_{i,UID}$ by first creating a normal key $D^*_{i,UID}$ and setting:

$$D_{i,UID} = (g_1^{\alpha_{i1}} H_1(UID)^{y_{i1}} h^{c_1 \cdot t_{i1}}, g_2^{\alpha_{i2}} H_2(UID)^{y_{i2}} h^{c_2 \cdot t_{i2}}, \cdots, g_m^{\alpha_{im}} H_m(UID)^{y_{im}} h^{c_m \cdot t_{im}}). \tag{10}$$

We note that when a semi-functional key of type 1 is used to decrypt a semi-functional ciphertext, the additional terms $\prod_{j=1}^{m} e(f,f)^{c_j \cdot u_{2,j,1}} = e(f,f)^{\sum_{j=1}^{m} c_j \cdot u_{2,j,1}}$ prevent decryption from succeeding except that when $\sum_{j=1}^{m} c_j \cdot u_{2,j,1} = 0$ (then we call it **nominally semi-function ciphertext**). When a semi-functional key of Type 2 is used to decrypt a semi-functional ciphertext, the additional terms $\prod_{j=1}^{m} e(h,h)^{c_j \cdot u_{3,j,1}} = e(h,h)^{\sum_{j=1}^{m} c_j \cdot u_{3,j,1}}$ prevent successful decryption.

Theorem 2. Our multi-authority CP-ABE with auxiliary inputs is $2^{-m^{\varepsilon}}$-AI-CPA secure if Assumptions 1 - 4 and the modified Goldreich-Levin theory hold.

Proof. We use a hybrid argument through a sequence of games to prove security. The first game $\text{Game}_{\text{real}}$ is an actual security game and the challenge access structure is denoted as $(\boldsymbol{A}^*, \rho)$ where ρ is injective. The second game Game_0 is the same as $\text{Game}_{\text{real}}$ except that the hash functions map identities *UID* to random elements of subgroup G_1 rather than G. The third game Game_1 is the same as Game_0 except that the adversary only obtains the semi-functional ciphertext. Let Q be the number of identities for which the adversary has queried the key for $D_{i,UID}$. After that, we define $\text{Game}_{q,1}$ and $\text{Game}_{q,2}$ for q = 0 to Q.

$\text{Game}_{q,1}$: It is the same as Game_1 except that for the first q-1 queried identities, the adversary receives semi-functional keys of type 2, for the q^{th} queried identity it receives a semi-functional key of type 1. The remaining keys are normal.

$\text{Game}_{q,2}$: It is the same as $\text{Game}_{q,1}$ except that the adversary receives a semi-functional key of type 2 for the q^{th} queried identities.

$\text{Game}_{\text{Final}}$: The ciphertext is a semi-functional one of a random message and all keys are type 2 semi-functional ones. Therefore, the adversary has no advantage in the game.

We need to prove that these games are indistinguishable in the lemmas as follows. Due to page limitation, we only give the detailed proofs of lemma 4 and lemma 5, and the proofs of other lemmas are relatively simple and can refer to the references [7, 14]. Let $\text{Game}_{\text{Real}}(\text{Adv}\mathcal{A})$, $\text{Game}_0(\text{Adv}\mathcal{A})$, $\text{Game}_{q,1}(\text{Adv}\mathcal{A})$, $\text{Game}_{q,2}(\text{Adv}\mathcal{A})$, $\text{Game}_{\text{Final}}(\text{Adv}\mathcal{A})$ denote the advantages of any PPT attacker $\mathcal{A}$ in the $\text{Game}_{\text{Real}}$, Game_0, $\text{Game}_{q,1}$, $\text{Game}_{q,2}$, $\text{Game}_{\text{Final}}$ respectively.

Lemma 1. Suppose there exists a polynomial time algorithm $\mathcal{A}$ such that $\text{Game}_{\text{Real}}(\text{Adv}\mathcal{A}) - \text{Game}_0(\text{Adv}\mathcal{A}) = \varepsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage ε in breaking Assumption 1.

Lemma 2. Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\text{Game}_0(\text{Adv}\mathcal{A}) - \text{Game}_1(\text{Adv}\mathcal{A}) = \varepsilon$. Then we can construct a polynomial time algorithm $\mathcal{B}$ with advantage negligibly close to ε in breaking Assumption 1.

Lemma 3. Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\text{Game}_{q\text{-}1,2}(\text{Adv}\mathcal{A}) - \text{Game}_{q,1}(\text{Adv}\mathcal{A}) = \varepsilon$. Then we can construct a PPT algorithm $\mathcal{B}$ with advantage negligibly close to ε in breaking Assumption 2.

Lemma 4. Suppose the attribute set of the q^{th} private key satisfies the challenge access policy, and the modified Goldreich-Levin theorem and Assumption 2 hold, then the PPT adversary $\mathcal{A}$ can distinguish a nominal semi-functional ciphertext from a truly semi-functional ciphertext with advantage negligibly close to ε.

Proof. We suppose that a challenger $\mathcal{C}$ of Goldreich-Levin theorem chooses one auxiliary function F and creates two vectors $\boldsymbol{d} = (d_1, d_2, \cdots, d_m) \in [0, \lambda]^m$, $\boldsymbol{\kappa} = (\kappa_1, \kappa_2, \cdots, \kappa_m) \in \text{GF}(p_2)^m$ and computes $F(\boldsymbol{d})$. $\mathcal{C}$ sends $\boldsymbol{\kappa}$, $F(\boldsymbol{d})$ and a random exponent $t \in \text{GF}(p_2)$ to a simulator $\mathcal{B}$. $\mathcal{B}$ will simulate $\text{Game}_{q,1}$ with an adversary $\mathcal{A}$. $\mathcal{A}$ specifies a set S' of corrupt authorities. Then $\mathcal{B}$ continues the game by running the setup algorithm for itself and giving $\mathcal{A}$ the pubic parameters. Since $\mathcal{B}$ knows the original master key and generators of all subgroups, it can make normal as well as semi-functional keys. Hence it can respond to $\mathcal{A}$'s queries of phase 1 by simply creating the queried keys.

When $\mathcal{A}$ submits the challenge access structure, $\mathcal{B}$ will not create the challenge key, but instead will encode the leakage $\mathcal{A}$ asks for on this decryption key in phase 2 as a single polynomial time computable function F. It can do this by fixing the values of all other keys and fixing all other variable invalued in the challenge key (more details on this below). Then $\mathcal{B}$ receive a sample $(\boldsymbol{\kappa}, F(\boldsymbol{d}), t)$, where $t = \langle \boldsymbol{\kappa}, \boldsymbol{d} \rangle$ or t is a random number. $\mathcal{B}$ will use $F(\boldsymbol{d})$ to answer all of $\mathcal{A}$'s leakage queries on the challenge key by implicitly defining the challenge key as follows. We let f denote a generator of G_2. $\mathcal{B}$ implicitly sets $c_j = (md_j\kappa_j - t)/m\kappa_j$ and $H_j(UID_q) = H_j^*(UID_q)\, f^{c_j}$ and $D_{i,UID} = (g_1^{\alpha_{i1}} H_1^*(UID)^{y_{i1}} f^{c_1 \cdot z_{i1}}, \cdots, g_m^{\alpha_{im}} H_m^*(UID)^{y_{im}} f^{c_m \cdot z_{im}})$.

At some point, $\mathcal{A}$ submits two messages M_0, M_1 to $\mathcal{B}$. $\mathcal{B}$ constructs the challenge ciphertext using $\boldsymbol{\kappa}$ such that $u_{2,j,1} = \kappa_j$. The remain parameters are chosen according to the EncryptSF algorithm. Now if $t = \langle \boldsymbol{\kappa}, \boldsymbol{d} \rangle$, then the challenge ciphertext is nominally semi-functional (and well distributed as such). If $t \neq \langle \boldsymbol{\kappa}, \boldsymbol{d} \rangle$, then the challenge ciphertext is truly semi-functional (and also well distributed).

It is clear that $\mathcal{B}$ can easily handle Phase 3 queries since the challenge key cannot be queried in here. When its attributes satisfy the challenge ciphertext's access structure. Hence $\mathcal{B}$ can use the output of $\mathcal{A}$ to gain a non-negligible advantage δ in distinguishing the distributions $(\boldsymbol{\kappa}, \boldsymbol{d}, \langle \boldsymbol{\kappa}, \boldsymbol{d} \rangle)$ and $(\boldsymbol{\kappa}, \boldsymbol{d}, t)$. Then $\mathcal{B}$ can invert the function $F(\boldsymbol{d})$ with advantage

$$\frac{\delta^3}{512mp_2^2} = \frac{p_2\delta^3}{512m} \cdot \frac{1}{p_2^3} > \frac{1}{p_2^3} = 2^{-m^{\varepsilon}}. \tag{11}$$

Thus $\mathcal{B}$ breaks the modified Goldreich-Levin theory.

However, if the attributes of the key do not satisfy the challenge access structure, the adversary can ask for the entire key to be revealed. Since the attributes of the key do not satisfy the challenge access structure, the row space $\boldsymbol{R} \subseteq Z_N^l$ is formed by rows of $\boldsymbol{A}$ whose attributes are in $B \subseteq S'$ and the rows whose attributes are queried by the adversary with identity UID_q, and this space cannot include the vector <1, 0, ⋯, 0>(we may assume this modulo p_2). So there is some vector $\boldsymbol{u}$ which is orthogonal to $\boldsymbol{R}$ modulo p_2 and not orthogonal to (1, 0, ⋯, 0). We can then write $\kappa_j cw = w_j' + a_j u'$ for some a_j modulo p_2 and w_j' in the span of the other basis vectors. We note that w_j' is uniformly distributed in this space and reveals no information about a_j. The value of the first coordinate of $\kappa_j cw$ modulo p_2 depends on the value of a_j, but the shares $\delta_{x,j}$ for $A_x \in B'$ (B' is the set of honest authority) contains no information about a_j. The only information that $\mathcal{A}$ receives about the value of a_j appears in the exponents of the form $\delta_{x,j} + \sum \gamma_{x,j} \cdot z_{\rho(x),j}$, where $z_{\rho(x),j}$ is a new random value, each time that appears nowhere else (recall that ρ is constrained to be injective) (we note that these $z_{\rho(x),j}$ values modulo p_2 do not occur in any keys for identities not equal to UID_q, since these keys are either normal or semi-functional of type 2 and hence do not have components in G_2). As long as $\gamma_{x,j}$ does not equal to 0 ($\gamma_{x,j} = 0$ with only negligible probability), this means that any value of $\delta_{x,j}$ can be explained by $z_{\rho(x),j}$ taking on a particular value. Since $z_{\rho(x),j}$ is uniformly random, this means that no information about the value of a_j modulo p_2 is revealed. Hence, the value being shared is information-theoretically hidden, and the $\delta_{x,j}$ is properly distributed in the adversary's view.

Since every $u_{2,j,1}$ is properly distributed in $\mathcal{A}$'s view, then $\sum c_j u_{2,j,1}$ is also properly distributed in $\mathcal{A}$'s view. $\mathcal{A}$ cannot tell whether it is a nominally semi-functional ciphertext. Though it is hidden from $\mathcal{A}$, the fact that we can only make $\delta_{x,j}$ shares of 0 is crucial here (*i.e.* the simulator can only make a nominally semi-functional ciphertext). If $\mathcal{B}$ tried to test the semi-functionality of the q^{th} key for itself by making a challenge ciphertext the key could decrypt, decryption would succeed regardless of the presence of G_2 components, since the $\delta_{x,j}$'s are shares of 0 and $\sum c_j u_{2,j,1} = 0$. Hence the simulator would not be able to tell whether the q^{th} key was semi-functional of Type 1 or normal.

In summary, when $T \in G_1$, $\mathcal{B}$ properly simulates $\text{Game}_{q-1,2}$. When $T \in G_{1,2}$, $\mathcal{B}$ properly simulates $\text{Game}_{q,1}$ with probability negligibly close to 1. Hence $\mathcal{B}$ can use $\mathcal{A}$ to obtain advantage negligibly close to ε in breaking Assumption 2.

Lemma 5. Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\text{Game}_{q,1}(\text{Adv}\mathcal{A}) - \text{Game}_{q,2}(\text{Adv}\mathcal{A}) = \varepsilon$. Then we can construct a PPT algorithm $\mathcal{B}$ with advantage ε in breaking Assumption 3.

Lemma 6. Suppose there exists a PPT algorithm $\mathcal{A}$ such that $\text{Game}_{q,2}(\text{Adv}\mathcal{A}) - \text{Game}_{\text{Final}}(\text{Adv}\mathcal{A}) = \varepsilon$. Then we can construct a PPT algorithm $\mathcal{B}$ with advantage ε in breaking Assumption 4.

Proof. $\mathcal{B}$ first receives $b_1, b_2, b_3, b_1^a, b_1^b b_3^b, b_1^c, b_1^{ac} b_3^d, T$. $\mathcal{B}$ will simulate either $\text{Game}_{q,2}$ or $\text{Game}_{\text{Final}}$ with $\mathcal{A}$, depending on the value of T. $\mathcal{B}$ randomly selects $\tau_1, \tau_2, \cdots, \tau_m \in Z_N$, and computes $g_1 = b_1^{\tau_1}$, $g_2 = b_1^{\tau_2}, \cdots, g_m = b_1^{\tau_m}$ as the public generators of G_1, and N as the group order. $\mathcal{A}$ specifies a set S' of corrupt authorities. For each attribute i belonging to a good authority, $\mathcal{B}$ chooses random vectors $\boldsymbol{\alpha_i}' = (\alpha_{i1}', \alpha_{i2}', \cdots, \alpha_{im}') \in Z_N^m$ and $\boldsymbol{y}_i' = (y_{i1}', y_{i2}', \cdots, y_{im}') \in Z_N^m$, and gives $\mathcal{A}$ the pubic parameters

$$\prod_{j=1}^{m} e(g_j, g_j)^{\alpha_{ij}} = \prod_{j=1}^{m} e(g_j, g_j)^{ab+\alpha_{ij}'} = \prod_{j=1}^{m} (e(b_1^a, b_1^b b_3^b)^{\tau_j^2} e(g_j, g_j)^{\alpha_{ij}'}), \tag{12}$$

$$\prod_{j=1}^{m} g_j^{y_{ij}} = \prod_{j=1}^{m} g_j^{a+y_{ij}'} = \prod_{j=1}^{m} ((b_1^a)^{\tau_j} g_j^{y_{ij}'}). \tag{13}$$

We note that this sets $\boldsymbol{\alpha_i} = (ab+\alpha_{i1}', ab+\alpha_{i2}', \cdots, ab+\alpha_{im}') \in Z_N^m$ and $\boldsymbol{y}_i = (ab+y_{i1}', ab+y_{i2}', \cdots, ab+y_{im}') \in Z_N^m$. When $\mathcal{A}$ queries the random oracle for $H_j(UID)$, $\mathcal{B}$ chooses random exponents $f_j, h_j \in Z_N$

and sets $H_j(UID) = ((b_1^b b_3^b)^{-1} b_1^{f_j} b_3^{h_j})^{\tau_j} = (b_1^{f_j - b})^{\tau_j} (b_3^{h_j - b})^{\tau_j}$. It stores this value. When $\mathcal{A}$ makes a key query (*i, UID*), $\mathcal{B}$ responds as follows. If $H_j(UID)$ has already been fixed, then $\mathcal{B}$ retrieves the stored value. Otherwise, $\mathcal{B}$ creates $H_j(UID)$ as above. $\mathcal{B}$ computes $D_{i, UID, j}$ as

$$g_j^{\alpha_{ij}} H_j^*(UID)^{y_{ij}} b_3^{(h_j - b)\tau_j y_{ij}'} = (b_1^{ab - \alpha_{ij}'})^{\tau_j} (b_1^{(f_j - b)(a + y_{ij}')})^{\tau_j} b_3^{(h_j - b)\tau_j y_{ij}'}$$
$$= (b_1^{\alpha_{ij}'} b_1^{f_j y_{ij}'} (b_1^a)^{f_j} b_3^{-b y_{ij}'} b_3^{(h_j - b) y_{ij}'})^{\tau_j} = (b_1^{\alpha_{ij}'} b_1^{f_j y_{ij}'} (b_1^a)^{f_j} (b_1^b b_3^b)^{-y_{ij}'} b_3^{h_j y_{ij}'})^{\tau_j} . \quad \textbf{(14)}$$

Notice that this sets $t_{ij} = \tau_j y_{ij}' \bmod p_3$ and the value is not correlated with the value of $y_{ij}' \bmod p_1$.

At some point, $\mathcal{A}$ gives $\mathcal{B}$ two messages M_0, M_1 and an access matrix ($\boldsymbol{A}$, ρ), before we create the challenge ciphertext. We first rewrite the ciphertext based on the generators b_1, b_2, b_3 as follows.

$$C_0 = M \cdot \prod_{i=1}^{m} e(g_j, g_j)^s = M \cdot e(b_1, b_1)^{s \sum \tau_j^2} , \quad \textbf{(15)}$$

$$C_{1,x,j} = e(g_j, g_j)^{\lambda_x} \cdot (\prod_{j=1}^{m} e(g_j, g_j)^{\alpha_{\rho(x),j}})^{r_{x,j}} = e(b_1, b_1)^{\tau_j^2 \lambda_x + \gamma_{x,j} \sum (\tau_j^2 \alpha_{\rho(x),j})} , \quad \textbf{(16)}$$

$$C_{2,x,j} = g_j^{r_{x,j}} f_j^{\gamma_{x,j}} h_j^{\psi_{x,j}} = b_1^{\tau_j r_{x,j}} b_1^{\tau_j \gamma_{x,j}} b_1^{\tau_j \psi_{x,j}} , \quad \textbf{(17)}$$

$$C_{3,x,j} = (\prod_{j=1}^{m} g_j^{y_{\rho(x),j}})^{r_{x,j}} g_j^{\omega_x} (\prod_{j=1}^{m} f_j^{z_{\rho(x),j}})^{\gamma_{x,j}} f^{\delta_{x,j}} (\prod_{j=1}^{m} h_j^{t_{\rho(x),j}})^{\psi_{x,j}} h^{\sigma_{x,j}}$$
$$= b_1^{\tau_j w_x + r_{x,j} \sum (\tau_j y_{\rho(x),j})} b_2^{\delta_{x,j} + \gamma_{x,j} \sum (\tau_j z_{\rho(x),j})} b_3^{\sigma_{x,j} + \psi_{x,j} \sum (\tau_j t_{\rho(x),j})}, \forall x, j \quad \textbf{(18)}$$

$\mathcal{A}$ additionally supplies $\mathcal{B}$ with public parameters $\prod_{j=1}^{m} e(g_j, g_j)^{\alpha_{ij}}$, $\prod_{j=1}^{m} g_j^{y_{ij}}$ for attributes *i* belonging to corrupt authorities which are included in the access matrix ($\boldsymbol{A}$, ρ).

$\mathcal{B}$ flips a random coin $\beta \in \{0, 1\}$, and encrypts M_β as follows. $\mathcal{B}$ sets: $C_0 = M_\beta T$. We think of this as setting $s = abc$. If $T = e(b_1, b_1)^{abc}$, then this will be an encryption of M_β. If T is random, this is will an encryption of a random message.

$\mathcal{B}$ chooses a random vector $\boldsymbol{u}_1$ with entries in Z_N, subject to the constraints that the first entry is 1 and $\boldsymbol{u}_1$ is orthogonal to all the rows in the set B' of good authorities (such a vector exists, otherwise the access matrix is illegal or a non-trivial factor of N can be found, violating our complexity assumptions, see [4]). We additionally choose a random vector $\boldsymbol{u}_2$ with entries in Z_N such that the first entry is 0 and the rest are randomly chosen. We define the vector $\boldsymbol{v} = abc\boldsymbol{u}_1 + \boldsymbol{u}_2$ (we note that this vector is uniformly random from $\mathcal{A}$'s perspective). We let $\lambda_x = A_x \cdot \boldsymbol{v} = abc\, A_x \boldsymbol{u}_1 + A_x \boldsymbol{u}_2$.

Since $\mathcal{B}$ cannot form the terms $e(b_1, b_1)^{\tau_j^2 abc A_x \cdot u_1}$ for rows $A_x \in B'$, it sets $r_{x,j} = -\frac{\tau_j^2 A_x \cdot u_1}{\sum \tau_j^2} c + r_{x,j}'$, where $r_{x,j}'$ is randomly chosen from Z_N. Then we have:

$$\tau_j^2 \lambda_x + r_{x,j} \sum (\tau_j^2 \alpha_{\rho(x),j}) = \tau_j^2 A_x \boldsymbol{u}_2 + (\sum \tau_j^2) r_{x,j}' ab + r_{x,j}' \sum (\tau_j^2 \alpha_{\rho(x),j}') - \frac{\tau_j^2 A_x \cdot u_1 \sum (\tau_j^2 \alpha_{\rho(x),j}')}{\sum \tau_j^2} c \quad \textbf{(19)}$$

This allows $\mathcal{B}$ to form $C_{1,x,j}$ for $A_x \in B'$ as:

$$C_{1,x,j} = e(b_1, b_1)^{\tau_j^2 A_x u_2 + r_{x,j} \sum (\tau_j^2 \alpha_{\rho(x),j})} e(b_1, b_1^c)^{\frac{\tau_j^2 A_x \cdot \boldsymbol{u}_1 \sum (\tau_j^2 \alpha_{\rho(x),j}')}{\sum \tau_j^2}} e(b_1^a, b_1^b b_3^b)^{r_{x,j} \sum \tau_j^2} , \quad \textbf{(20)}$$

For rows $A_x \in B'$ corresponding to corrupt authorities, $\mathcal{B}$ chooses $r_x \in Z_N$ randomly and sets:

$$C_{1,x,j} = e(b_1, b_1)^{\tau_j^2 A_x u_2 + r_{x,j} \sum(\tau_j^2 \alpha_{\rho(x),j})} e(b_1^a, b_1^b b_3^b)^{r_{x,j} \sum \tau_j^2}, \quad (21)$$

For rows $A_x \in B$, $\mathcal{B}$ can form $C_{2,x}$ by choosing a random value $r_{x,j} \in Z_N$ randomly and setting:

$$C_{2,x,j} = (b_1^c)^{-\frac{\tau_j^3 A_x \cdot \boldsymbol{u}_1}{\sum \tau_j^2}} b_1^{\tau_j r'_{x,j}} (b_2 b_3)^{\gamma_{x,j}}, \quad (22)$$

We note that the values of $\gamma_{x,j}$ modulo p_2 and p_3 are uncorrelated, so this is properly distributed. For rows $A_x \in B$, $\mathcal{B}$ can simply compute $C_{2,x,j} = b_1^{\tau_j r_{x,j}}$.

Now $\mathcal{B}$ chooses a random vector $\boldsymbol{w}$ with first entry equal to 0 and other entries randomly chosen from Z_N, and m random vectors $\boldsymbol{u}_{3,1}, \boldsymbol{u}_{3,2}, \cdots \boldsymbol{u}_{3,m}$ whose entries are all randomly chosen from Z_N. We let $\omega_x = A_x \cdot \boldsymbol{w}$ and $\delta_{x,j} = A_x \cdot u_{3,j}$. For rows $A_x \in B'$, we note that

$$\gamma_{x,j} \sum(\tau_j y_{\rho(x),j}) = -ac \frac{A_x \cdot \boldsymbol{u}_1 \tau_j^2 \sum \tau_j}{\sum \tau_j^2} + a\gamma'_{x,j} \sum \tau_j - c \frac{\tau_j^2 A_x \cdot \boldsymbol{u}_1 \sum(\tau_j^2 y'_{\rho(x),j})}{\sum \tau_j^2} + \gamma'_{x,j} \sum(\tau_j y'_{\rho(x),j}). \quad (23)$$

So $\mathcal{B}$ can form $C_{3,x,j}$ as:

$$C_{3,x,j} = b_1^{\tau_j w_x} b_1^{\gamma'_{x,j} \sum(\tau_j y'_{\rho(x),j})} (b_1^a)^{\gamma'_{x,j} \sum \tau_j} (b_1^c)^{\frac{\tau_j^2 A_x \cdot \boldsymbol{u}_1 \sum(\tau_j^2 y'_{\rho(x),j})}{\sum \tau_j^2}} (b_1^{ac} b_3^d)^{\frac{\tau_j^2 A_x \cdot \boldsymbol{u}_1 \sum \tau_j}{\sum \tau_j^2}} (b_2 b_3)^{\delta_{x,j} + \gamma_{x,j} \sum(\tau_j y'_{\rho(x),j})}. \quad (24)$$

(This is consistent with $t_{\rho(x),j}$ being congruent to $y_{\rho(x),j}'$ modulo p_3 in the keys.) We note that the sharing vector in subgroups G_2 and G_3 is $u_{3,j}$ and $u_{3,j}$- $(\boldsymbol{u}_1 d \tau_j^2 \Sigma \tau_j) / \Sigma \tau_j^2$, which is random modulo p_2 and modulo p_3. For rows $A_x \in B'$, $\mathcal{B}$ sets:

$$C_{3,x,j} = b_1^{\tau_j w_x} b_1^{\gamma_{x,j} \sum(\tau_j y'_{\rho(x),j})} (b_1^a)^{\sum \tau_j \gamma_{x,j}} (b_2 b_3)^{A_x \cdot \boldsymbol{u}_{3,j}}, \quad (25)$$

The sharing vector is consistent here because u_1 is orthogonal to all of these rows A_x. This is a properly distributed semi-functional ciphertext with $s = abc$. If $T = e(b_1, b_1)^{abc}$, this is a semi-functional encryption of M_β, and $\mathcal{B}$ has simulated $\text{Game}_{Q,2}$. If T is random, then this is a semi-functional encryption of a random message, so $\mathcal{B}$ has simulated Game_{Final}. Hence, $\mathcal{B}$ can use $\mathcal{A}$ to obtain advantage ε in breaking Assumption 4.

In conclusion, these games are proved to be indistinguishable from Lemma 1 to Lemma 6, and then the attacker has no advantage in the real system to break its security. Therefore, the theorem 2 holds.

5.2 Performance Comparisons

This section shows the performance comparisons with Lewko et al. scheme [14], Wang et al. scheme [15] and our scheme. The three schemes are all leakage-resilient CP-ABE schemes. Let P_c denote a pairing cost, E_c denote an exponent cost, M_c denote a multiplication cost. We assume that the LSSS access matrix $\boldsymbol{A}$ is $n \times l$. Let ϖ denote the leakage parameter, ξ denote the allowable leakage probability parameter.

Table 1. Performance comparisons

Schemes	Lewko [14]	Wang [15]	Our scheme
Encryption cost	$2(\varpi+2n)\ M_c$	$2(1+m+2n)\ E_c$	$(1+3nm)\ E_c$
Decryption cost	$(\varpi+2n+1)\ P_c$	$2(m+2\|I\|)\ P_c$	$\|I\|(m+1)\ P_c$
Leakage bound	$2+(\varpi-1-2\xi)\log p_2$	No	No
Leakage model	Bounded leakage	Auxiliary inputs	Auxiliary inputs
Supporting multi-authority	No	No	Yes

Table 1 shows that the computational cost of [14] is primarily dependent on the leakage parameter ϖ, while the computational costs of [15] and our scheme are primarily dependent on the number m of pieces. Compared with Lewko et al. scheme [14], our scheme allows the unbounded leakage of user's private

key and may support multi-authority applications. Compared with Wang et al. scheme [15], our scheme may support multi-authority applications. Although our scheme's encryption and decryption costs are moderately higher than two works [14-15], our scheme may support the unbounded leakage and multi-authority applications simultaneously.

6 Conclusions

We first propose the multi-authority CP-ABE resilient against auxiliary-input leakage, which remains fully secure even if the attacker gets the leakage of the private key with any auxiliary input function. Our scheme not only combines the benefits of auxiliary input leakage resilience and dual system encryption, but also achieves numeric unbounded leakage on the attribute-based private keys of users. We proved that it is fully secure under the modified Goldreich-Levin theorem and the subgroup decision problem assumptions. Compared with the relative leakage-resilient ABE schemes [14-15], our scheme may achieve numeric unbounded leakage on the attribute-based private keys of users and support multi-authority applications simultaneously. Therefore, our scheme can be deployed in a large scale distributed system with side-channel attacks such as public cloud computing.

However, our scheme cannot allow the master secret key leakage and continual leakage of secret key, we regard it as future research work. In this work, the master key and the user private key have different structures. To allow attacker to obtain the leakage of master secret key, it is critical to make the master key and user private key have the same structure. In our scheme, the user private key is tied with his identity and not tied with a random component. The key factor of continuous leakage is to have some random component to tie the secret key so that the secret key can be updated periodically. In our future works, to resist the largest possible class of potential attackers, we will modify this work to construct a multi-authority ABE with continual auxiliary-input leakage which can allow master key leakage and continuous leakage.

Acknowledgements

We are grateful to the anonymous reviewers for their invaluable comments. This work is supported by Jiangsu Overseas Visiting Scholar Program for University Prominent Young and Middle-aged Teachers and Presidents, the National Natural Science Foundation of China (No. 61402244, No. 11371207, No. 61762044), Nantong City Application Basic Research Project (No. GY12017024), the Zhejiang Natural Science Foundation (LY15F020010).

References

[1] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: Proc. 24th Annual International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT'05), 2005.

[2] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in Proc. IEEE Symp. Security and Privacy, 2007.

[3] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proc. 13th ACM Conference on Computer and Communication Security, 2006.

[4] M. Chase, Multi-authority attribute based encryption, in: Proc. 4th Theory of Cryptography Conference, 2007.

[5] M. Chase, S. Chow, Improving privacy and security in multi-authority attribute-based encryption, in: Proc. 16th ACM Conference on Computer and Communications Security, 2009.

[6] N. Gorasia, R.R. Srikanth, N. Doshi, J. Rupareliya, Improving security in multi authority attribute based encryption with fast decryption, Procedia Computer Science 79(2016) 632-639.

[7] A. Lewko, B. Waters, Decentralizing attribute-based encryption, in: Proc. Advances in Cryptology-EUROCRYPT 2011, 2011.

[8] W. Wang, F. Qi, X. Wu, Z. Tang, Distributed multi-authority attribute-based encryption scheme for friend discovery in mobile social networks, Procedia Computer Science 80(2016) 617-626.

[9] P. C. Kocher, Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems, in: Proc. the 16th Annual International Cryptology Conference, 1996.

[10] P. C. Kocher, J. Jaffe, and B. Jun, Differential power analysis, in: Proc. the 19th international Cryptology Conference, 1999.

[11] A. Akavia, S. Goldwasser, V. Vaikuntanathan, Simultaneous hardcore bits and cryptography against memory attacks, in: Proc. the 29th International Cryptology Conference, 2009.

[12] J. Alwen, Y. Dodis, M. Naor, G. Segev, S. Walfish, D. Wichs, Public-key encryption in the bounded-retrieval model, in: Proc. the 29th annual international Conference on the Theory and Application of Cryptographic Techniques, 2010.

[13] Y. Dodis, S. Goldwasser, Y.T. Kalai, O. Peikert, V. Vaikuntanethan, Public-key encryption schemes with auxiliary inputs, in: Proc. the Theory of Cryptography Conference, 2010.

[14] A. Lewko, Y. Rouselakis, B. Waters, Achieving leakage resilience through dual system encryption, in: Proc. 8th IACR Theory of Cryptography Conference, 2011.

[15] Z. Wang, S.M. Yiu, Attribute-based encryption resilient to auxiliary input, in: Proc. ProvSec 2015, 2015.

[16] H. Ma, G. Zeng, Z. Bao, J. Chen, J. Wang, Z.J. Wang, Attribute-based encryption scheme resilient against continuous auxiliary-inputs leakage, Journal of Computer Research and Development 53(8)(2016) 1867-1878.

[17] M. Zhang, W. Shi, C. Wang, Z. Chen, Y. Mu, Leakage-resilience attribute-based encryption with fast decryption: models, analysis and constructions, in: Proc. ISPEC 2013, 2013.

[18] J. Li, Q. Yu, Y. Zhang, J. Shen, Key-policy attribute-based encryption against continual auxiliary input leakage, Information Sciences, 470, 10.1016/j.ins.2018.07.077.

[19] L. Zhang, J. Zhang, Y. Hu, Attribute-based encryption resilient to continual auxiliary leakage with constant size ciphertexts, Journal of China Universities of Posts and Telecommunications 23(3)(2016) 18-28.

[20] J. Li, Q. Huang, X. Chen, S. SM Chow, D.S. Wong, D. Xie, Multi-authority ciphertext-policy attribute-based encryption with accountability, In: Proc. the 6th ACM Symposium on Information, Computer and Communications Security, 2011.

[21] H. Ma, G. Zeng, Z. Wang, J. Xu, Fully secure multi-authority attribute-based traitor tracing, Journal of Computational Information Systems 9(7)(2013) 2793-2800.

[22] S. Micali, L. Reyzin, Physically observable cryptography, in: Proc. TCC, 2004.

[23] J. Katz, V. Vaikuntanathan. Signature schemes with bounded leakage resilience, in: Proc. ASIACRYPT, 2009.

[24] J. Alwen, Y. Dodis, D. Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model, in: CRYPTO, 2009.

[25] Y. Dodis, K. Haralambiev, A. Lopez-Alt, D. Wichs, Cryptography against continuous memory attacks, in: Proc. FOCS, 2010.

[26] Z. Brakerski, Y.T. Kalai, J. Katz, V. Vaikuntanathan, Overcoming the hole in the bucket: Publickey cryptography resilient to continual memory leakage, in: Proc. FOCS, 2010.

[27] T.H. Yuen, S.S. M. Chow, Y. Zhang, S.M. Yiu, Identity-based encryption resilient to continual auxiliary leakage, in: Proc. EUROCRYPT 2012, 2012.