Attribute-Based Encryption Resilient to Auxiliary Input

Zhiwei Wang[1,2(✉)] and Siu Ming Yiu[2]

[1] College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, Jiangsu, China
zhwwang@njupt.edu.cn
[2] University of Hong Kong, Pokfulam, Hong Kong

Abstract. The *auxiliary input* model defines a class of computationally uninvertible function families $\mathcal{F}$ to simulate a large class of leakage. Such a function $f \in \mathcal{F}$ can information-theoretically reveal the entire secret key SK, but it is still computationally infeasible to recover SK from $f(SK)$. That means SK can be used for multiple tasks, since SK doesn't need to be continually refreshed. We propose the first CP-ABE scheme based on linear secret sharing schemes, that can tolerate leakage on master key and attribute-based secret keys with *auxiliary input*(AI). For the security proof of our scheme, we present three modified assumptions in composite order bilinear groups, and prove their hardness. Under these modified assumptions, our scheme can be proved AI-CPA secure in the standard model. Finally, we devise a key-policy ABE scheme also resilient to *auxiliary input*.

Keywords: Leakage resilience · Attribute-based encryption · Auxiliary input · Linear secret sharing scheme

1 Introduction

With the development of cloud computing, there is a trend for users to store their data on the cloud server. It is inefficient to distribute these encrypted data to a specific set of users in traditional cryptosystems, e.g., PKI, ID-based cryptosystem, since the cipher-text size and computational cost of encryption/decryption algorithms are linear with the number of receivers. For this reason, Sahai and Waters [1] firstly proposed the concept of attribute-based encryption. In attribute-based encryption, cipher-texts and keys are associated with sets of attributes and access structure over attributes. Only when the attributes of the cipher-text match those of the users' key, the corresponding cipher-text can be decrypted. There are two kinds of ABE systems: The first one is ciphertext-policy ABE (CP-ABE), where cipher-texts are associated with access structures and keys are associated with sets of attributes; the second one is key-policy ABE (KP-ABE), where keys are associated with access structure and cipher-texts are associated with sets of attributes.

How to achieve a more expressive access policy over many attributes is an important problem in ABE. Sahai and Waters's [1] scheme was limited to specify

M.-H. Au and A. Miyaji (Eds.): ProvSec 2015, LNCS 9451, pp. 371–390, 2015.
DOI: 10.1007/978-3-319-26059-4_21

as threshold access policies with one threshold gate. After then, Lewko et al. [5] used monotone span programs (MSPs) as access structure to devise a CP-ABE and a KP-ABE, which are proved secure in composite bilinear groups. However, their schemes are very inefficient, since the length of cipher-texts and keys, and the number of pairings in decryption are all polynomial in the size of MSPs. In order to improve the efficiency, some ABE systems make use of linear secret sharing scheme (LSSS) or boolean formulas as access structure. Waters [10] employed LSSS matrix as access structure to realize CP-ABE under concrete and noninteractive cryptographic assumptions. In [6], Goyal et al. provided a mapping from a universal access tree to formulas consisting of threshold gates. They used this technique to construct a bounded CP-ABE scheme. There is a close relation between LSSS and MSP access structure. Beimel et al. [7] proved that the existence of a LSSS for a specific MSP access structure is equivalent to a smallest MSP. Pandit et al. [8] used minimal sets to realize the smallest MSP for describing general access structure in ABE systems. Recently, Zhang et al. [12] proposed a CP-ABE and a KP-ABE resilient to continual leakage by minimal sets.

In practice, many cryptosystems are difficult to avoid the side-channel attacks, which allow attackers to learn partial information about secret by observing physical properties of a cryptographic execution such as timing, power assumption, temperature, radiation, etc. [14–18]. The concept of leakage resilient cryptography has been proposed, which has led to construction of many cryptographic primitives which can be proved secure even against adversaries who can obtain partial information of secret keys and other initial state. Leakage resilience has been studied in many previous work under a variety of leakage models. We review these leakage models as follows:

Exposure-resilient: This model addressed adversaries who could learn a subset of the bits of the secret key or internal state [19,20].

Only computation leaks information: In this model, it is assumed that leakage occurs every time the device performs a computation, but any part of the memory not involved in computation does not leak [21,22].

Bounded retrieval model: In this model, the total number of bits leaked over the lifetime of system is significantly less than the bit-length of the key, and hope the attack is detected and stopped before the whole secret is leaked. This model has been employed successfully in many constructions of cryptographic primitives [23–25].

Continual leakage model: In this model, it is assumed the leakage between consecutive updates is bounded in term of a fraction of the secret key size, and the secret key should be refreshed continually. There is no leakage during the update process. Dodis et al. [26] constructed one-way relations, signatures, identification schemes, and authenticated key agreement protocols resilient to continual leakage. Lewko et al. [27], proposed fully secure IBE, HIBE, ABE systems which are be realized as resilience against continual leakage. Zhang et al. [12] also proposed a CP-ABE system and a KP-ABE system resilient to continual leakage.

Auxiliary input model: Auxiliary input model is developed from the *relative leakage* model [14], which allows any uninvertible function f that no PPT adversary can compute the actual pre-image with non-negligible probability[1]. That is to say, although such a function information-theoretically reveals the entire secret key SK, it still computationally infeasible to recover SK from $f(SK)$. If an encryption scheme is secure w.r.t. any auxiliary input, then user's secret and public key pair can be used for multiple tasks. Dodis et al. [13] firstly introduced the notion of *auxiliary input*, and proposed the public key encryption schemes in this model. Yuen et al. [3] proposed the first IBE scheme that is proved secure even when the adversary is equipped with auxiliary input. In [3], they also propose the model of *continual auxiliary leakage* that combines the concepts of auxiliary inputs with continual memory leakage.

Recently, Waters [9] introduced a new technique for security proof called *dual system encryption*, in which there are two kinds of keys and cipher-texts: *normal* and *semi-functional*[2]. Normal keys can decrypt both forms of cipher-texts, while semi-functional keys can only decrypt normal cipher-texts. In the real game, keys and cipher-texts are all normal, but they will be transformed into semi-functional one by one in the security proof. We must prove that the adversary cannot distinguish these transformations. In the final game, all keys and cipher-texts are semi-functional, which cannot be decrypted correctly. Lewko et al. [27] showed that the technique of dual system encryption and leakage resilience are highly compatible, their combination not only improves the leakage tolerance of cryptographic primitives, but also no sacrifices of efficiency.

Our Contribution. In this work, we propose the first CP-ABE scheme that is secure in presence of *auxiliary input*. After extension, our scheme can be transformed to a CP-ABE scheme resilient to *continual auxiliary leakage*. Our construction is based on Waters' most efficient construction of CP-ABE [10]. Our scheme in Sect. 4 preserves the nice features of Waters' scheme: security in the standard model, and based on static assumptions. In order to resist the leakage in form of auxiliary input and continual auxiliary leakage, we use the *GL Theorem for Large Fields*. The key point for using the *GL Theorem for Large Fields* is how to split the secret key into m pieces, since the *GL Theorem for Large Fields* states that if the pieces of secret key α_i belongs to a subgroup H of $\mathbb{Z}_{p_1}$(p_1 is a λ-bit prime.), then the running time of inverter is $poly(|H|)$. Thus, if H is a large field, and close to $\mathbb{Z}_{p_1}$(λ is a security parameter.), then the running time is close to $poly(2^{\lambda})$, which is undesirable for the inverter. Our scheme also can be extended to an ABE scheme resilient to *continual auxiliary leakage*, if it doesn't allow leakage during the setup phase.

Lewko et al. [4] proposed three static assumptions in composite order groups, which has been used in many constructions [3,12,27]. However, they cannot

[1] "non-negligible probability" means that the probability cannot be ignored.

[2] The definitions of *normal* and *semi-functional* are only for proof, and they are not concerned with construction.

be directly used in the security proof of our constructions. We propose three modified assumptions and prove their hardness by using the two theorems in [9]. Another technical difficulty in our security proof is the form of attribute. If we use 2-SDP assumption to prove the Lemma 1, each attribute should be a integer number in $\mathbb{Z}_N$. Thus, we must pre-define an injective map from the attributes space to $\mathbb{Z}_N$. Since attributes space can be public, this map also can be public, and has no impact to the security level of ABE scheme.

Organization. In Sect. 2, we propose three modified complexity assumptions, and their proofs are provided in Appendix A. In Sect. 3, we provide the security model of CP-ABE resilient to auxiliary input. In Sect. 4, we devise a concrete CP-ABE scheme resilient to auxiliary input based on LSSS scheme. In Sect. 5, we prove our scheme by using the technique of dual system encryption. In Sect. 6, we design a KP-ABE scheme resilient to auxiliary input. In Sect. 7, we conclude our paper.

2 Background

In this section, we firstly give the definitions and proofs to our modified hard assumptions. Secondly, we provide the formal definitions for access structures and Linear Secret Sharing Scheme (LSSS).

2.1 Hardness Assumptions

Bilinear groups of composite order are groups introduced by [2], where the group order is product of two or more distinct primes. In our construction, we use the group order of $N = p_1p_2p_3$, where p_1, p_2, p_3 are three distinct prime numbers. We denote this group as $\mathbb{G}$, and admit an efficient bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, where $\mathbb{G}_T$'s order is the same as $\mathbb{G}$'s. Any element of $\mathbb{G}$ can be denoted as $g_1^{a_1} g_2^{a_2} g_3^{a_3}$, where g_i is the generator of subgroup $\mathbb{G}_{p_i}$. Each $\mathbb{G}_{p_i}$ has the order p_i, and $a_i \in \mathbb{Z}_{p_i}$. We denote $\mathbb{G}_{p_ip_j}$ as the subgroup of order p_ip_j in $\mathbb{G}$. For all $T \in \mathbb{G}_{p_ip_j}$, T can be defined as the product of an element in $\mathbb{G}_{p_i}$ and an element in $\mathbb{G}_{p_j}$. For all $v \in \mathbb{G}_{p_i}$ and $w \in \mathbb{G}_{p_j}$, $\hat{e}(v, w) = 1$ if $i \neq j$. The following three hardness assumptions, which have been analyzed in [4,5], have been used in many constructions [3,12,27].

Definition 1 (1-SDP assumption). *Given* $\Theta = (N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, \hat{e})$, *if for all PPT algorithm* $\mathfrak{A}$, *there exists a negligible probability* ϵ *such that*

$$|Pr[\mathfrak{A}(\Theta, g_1, X_3, T_0) = 1] - Pr[\mathfrak{A}(\Theta, g_1, X_3, T_1) = 1]| \leq \epsilon,$$

where the probabilities are taken over the choice of $g_1 \in \mathbb{G}_{p_1}, X_3 \in \mathbb{G}_{p_3}, T_0 \in \mathbb{G}_{p_1p_2}, T_1 \in \mathbb{G}_{p_1}$.

Definition 2 (2-SDP assumption). *Given* $\Theta = (N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, \hat{e})$, *if for all PPT algorithm* $\mathfrak{A}$, *there exists a negligible probability* ϵ *such that*

$$|Pr[\mathfrak{A}(\Theta, g_1, X_1X_2, X_3, Y_2Y_3, T_0) = 1] - Pr[\mathfrak{A}(\Theta, g_1, X_1X_2, X_3, Y_2Y_3, T_1) = 1]| \leq \epsilon,$$

where the probabilities are taken over the choice of $g_1 \in \mathbb{G}_{p_1}, X_2, Y_2 \in \mathbb{G}_{p_2}, X_3, Y_3 \in \mathbb{G}_{p_3}, T_0 \in \mathbb{G}_{p_1 p_3}, T_1 \in \mathbb{G}$.

Definition 3 (BSDP assumption). *Given* $\Theta = (N = p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, \hat{e})$*, if for all PPT algorithm* $\mathfrak{A}$*, there exists a negligible probability* ϵ *such that*

$$|Pr[\mathfrak{A}(\Theta, g_1, g_1^{\alpha} X_2, X_3, g_1^{s} Y_2, Z_2, T_0) = 1] - Pr[\mathfrak{A}(\Theta, g_1, g_1^{\alpha} X_2, X_3, g_1^{s} Y_2, Z_2, T_1) = 1]| \leq \epsilon,$$

where the probabilities are taken over the choice of $s, \alpha \in \mathbb{Z}_N, g_1 \in \mathbb{G}_{p_1}, X_2, Y_2, Z_2 \in \mathbb{G}_{p_2}, X_3 \in \mathbb{G}_{p_3}, T_0 = \hat{e}(g_1^{\alpha}, g_1^{s}), T_1 \in \mathbb{G}_T$.

However, our construction in Sect. 4 should be proved secure under the following three modified assumptions. Let $[m]$ denote $\{1, \cdots, m\}$.

Definition 4 (modified 1-SDP assumption). *Given* $\Theta = (N = p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, \hat{e})$*, if for all PPT algorithm* $\mathfrak{A}$*, there exists negligible probabilities* $\epsilon_1, \cdots, \epsilon_m$ *such that*

$$\begin{aligned} |Pr[\mathfrak{A}(\Theta, g_1, X_3, T_{01}) = 1] - Pr[\mathfrak{A}(\Theta, g_1, X_3, T_{11}) = 1]| &\leq \epsilon_1, \\ &\vdots \\ |Pr[\mathfrak{A}(\Theta, g_1, X_3, T_{0m}) = 1] - Pr[\mathfrak{A}(\Theta, g_1, X_3, T_{1m}) = 1]| &\leq \epsilon_m, \end{aligned}$$

where the probabilities are taken over the choice of $g_1 \in \mathbb{G}_{p_1}, X_3 \in \mathbb{G}_{p_3}, T_{0i} \in \mathbb{G}_{p_1 p_2}, T_{1i} \in \mathbb{G}_{p_1}$.

Definition 5 (modified 2-SDP assumption). *Given* $\Theta = (N = p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, \hat{e})$*, if for all PPT algorithm* $\mathfrak{A}$*, there exists a negligible probability* ϵ *such that*

$$\begin{aligned} &|Pr[\mathfrak{A}(\Theta, g_1, (X_{1i} X_{2i})_{i \in [m]}, X_3, Y_2 Y_3, T_0) = 1] \\ -&Pr[\mathfrak{A}(\Theta, g_1, (X_{1i} X_{2i})_{i \in [m]}, X_3, Y_2 Y_3, T_1) = 1]| \leq \epsilon, \end{aligned}$$

where the probabilities are taken over the choice of $g_1 \in \mathbb{G}_{p_1}, X_{2i}, Y_2 \in \mathbb{G}_{p_2}, X_3, Y_3 \in \mathbb{G}_{p_3}, T_0 \in \mathbb{G}_{p_1 p_3}, T_1 \in \mathbb{G}$.

Definition 6 (modified BSDP assumption). *Given* $\Theta = (N = p_1 p_2 p_3, \mathbb{G}, \mathbb{G}_T, \hat{e})$*, if for all PPT algorithm* $\mathfrak{A}$*, there exists a negligible probability* ϵ *such that*

$$\begin{aligned} |Pr[\mathfrak{A}\ &(\Theta, g_1, (g_1^{1/b_i})_{i \in [m]}, (B_i^{\alpha_i} X_2)_{i \in [m]}, X_3, (B_i^{s_i} Y_2)_{i \in [m]}, Z_2, T_0) = 1] \\ -&Pr[\mathfrak{A}(\Theta, g_1, (g_1^{1/b_i})_{i \in [m]}, (B_i^{\alpha_i} X_2)_{i \in [m]}, X_3, (B_i^{s_i} Y_2)_{i \in [m]}, Z_2, T_1) = 1]| \leq \epsilon, \end{aligned}$$

where the probabilities are taken over the choice of $s_i, \alpha_i, b_i \in \mathbb{Z}_N, g_1, B_i = g_1^{b_i} \in \mathbb{G}_{p_1}, X_2, Y_2, Z_2 \in \mathbb{G}_{p_2}, X_3 \in \mathbb{G}_{p_3}, T_0 = \prod_{i=1}^{m} \hat{e}(g_1, B_i)^{\alpha_i s_i}, T_1 \in \mathbb{G}_T$.

We prove the hardness of three modified assumptions in **Appendix** A.

2.2 Access Structure and Linear Secret Sharing Scheme

We adapt our definitions which are given by [11]. However, the role of parties is taken by the attributes in our definitions.

Definition 7 (Access Structure). *Let $\{S_1, \cdots, S_n\}$ be a set of attributes. An authorized collection $\mathbb{A} \subset 2^{\{S_1,\cdots,S_n\}}$ is monotone, if $\forall B, C$, $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. A monotone access structure is a monotone collection $\mathbb{A}$ of non-empty of subsets of $\{S_1, \cdots, S_n\}$. The sets not in $\mathbb{A}$ are called the unauthorized sets.*

Definition 8 (Linear Secret Sharing Scheme (LSSS)). *A secret sharing scheme Π over a set of attributes $\mathcal{S}$ is called linear on the two conditions that: 1) The shares for each attributes form a vector from $\mathbb{Z}_N$. 2) There exists an $l \times n$ matrix $\mathcal{A}$ called sharing-generating matrix for Π. For all $i = 1, \cdots, l$, the function ρ maps the i-th row of $\mathcal{A}$ to an attribute labeling $\rho(i)$. Then, we selects a random column vector $v = (\mu, r_2, \cdots, r_n)$ where $\mu \in \mathbb{Z}_N$ is the secret to be shared, and $\mathcal{A}v$ is the vector of l shares of the secret μ according to Π. The share $(\mathcal{A}v)_i$ belongs to the attribute $\rho(i)$.*

From the discussion in [11], each LSSS scheme Π for the access structure $\mathbb{A}$ has a property of *linear reconstruction*. Let $\mathcal{C} \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, \cdots, l\}$ be defined as $I = \{i : \rho(i) \in \mathcal{C}\}$. Then, there exists constant $\{\omega_i \in \mathbb{Z}_N\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any μ in Π, then $\sum_{i \in I} \omega_i \lambda_i = \mu$. These $\{\omega_i\}$ can be found in polynomial time in the size of matrix $\mathcal{A}$.

3 Attribute Based Encryption with Auxiliary Inputs

In this section, we give the security model of cipher-text-policy ABE resilient to auxiliary input (AI-CP-ABE), where the access structure is monotonic. In Sect. 6, we will provide the concrete scheme of key-policy ABE resilient to auxiliary input (AI-KP-ABE).

A CP-ABE for a general monotone access structure $\mathbb{A}$ over the monotone attribute universe space Σ is composed of four probabilistic polynomial time algorithms:

1. **Setup**$(1^\lambda, \Sigma)$: The setup algorithm takes as input a security parameter λ and an attribute set Σ, and outputs system public key MPK and master key MSK.
2. **KeyGen**$(MSK, \mathbb{S})$: This algorithm takes as input an attribute set $\mathbb{S}$, and the master secret key MSK, and outputs a private key $SK_{\mathbb{S}}$.
3. **Encrypt**$(M, \mathbb{A})$: The encryption algorithm takes as input a monotone access structure $\mathbb{A}$ and a message M, and outputs a cipher-text CT.
4. **Decrypt**(CT, SK): This algorithm takes as input a cipher-text CT for an access structure $\mathbb{A}$ and a private key SK for a set $\mathbb{S}$, and outputs M if and only if the attribute set $\mathbb{S}$ satisfies the monotone access structure $\mathbb{A}$.

Let Σ and $\mathcal{M}$ be the monotone attribute space and the message space respectively. $\forall M \in \mathcal{M}$, $\forall \mathbb{A}$[3] $\in 2^\Sigma$ and $\forall \mathbb{S} \in \mathbb{A}$, $M \leftarrow$ Decrypt$(SK,$

[3] The access structure $\mathbb{A}$ is a monotone collection of non-empty of subsets of Σ.

Encrypt$(MPK, M, \mathbb{A})$), where $(MPK, MSK) \leftarrow \text{Setup}(1^\lambda, \Sigma)$, $SK \leftarrow KeyGen(MSK, \mathbb{S})$.

3.1 Security Model of AI-CP-ABE

In this section, we provide the security model of ciphertext-policy attribute based encryption for semantic security with leakage in form of auxiliary input (AI-CP-ABE). Let $\mathcal{F}$ denote a polynomial time computable function family. We define the security model by an indistinguishable game between a challenger $\mathfrak{C}$ and an adversary $\mathfrak{A}$. In order to record the queried and leaked keys, we set two empty lists: $\mathfrak{R} = \langle \bar{j}, \mathbb{S} \rangle$ and $\mathfrak{Q} = \langle \bar{j}, \mathbb{S}, SK_{\mathbb{S}} \rangle$, where $\bar{j}$ is a handle index[4].

Setup. The challenger $\mathfrak{C}$ runs the Setup algorithm to generate MPK and MSK, and sends MPK to $\mathfrak{A}$.

Query 1. The adversary $\mathfrak{A}$ can perform the following queries:

- **Key extraction query($\mathcal{Q}_E$):** When $\mathfrak{A}$ makes a key extraction query on an attribute set $\mathbb{S} \subset \Sigma$, $\mathfrak{C}$ checks the list $\mathfrak{Q}$ for the tuple with the form $\langle \bar{j}, \mathbb{S}, SK_{\mathbb{S}} \rangle$. If there is no such tuple, then $\bar{j}$ is set to 1, and $\mathfrak{C}$ answers $SK_{\mathbb{S}} \leftarrow KeyGen(MSK, \mathbb{S})$. Then, $\mathfrak{C}$ puts $\langle \bar{j}, \mathbb{S}, SK_{\mathbb{S}} \rangle$ into the list $\mathfrak{Q}$. Otherwise, $\mathfrak{C}$ returns $SK_{\mathbb{S}}$ from the tuple $\langle \bar{j}, \mathbb{S}, SK_{\mathbb{S}} \rangle$, and set $\bar{j} = \bar{j} + 1$.
- **Key leakage query($\mathcal{Q}_L$):** When $\mathfrak{A}$ makes a key leakage query on an attribute set $\mathbb{S} \subset \Sigma$ with a function $f \in \mathcal{F}$, $\mathfrak{C}$ returns $f(MSK, \mathfrak{Q}, MPK, \mathbb{S})$.
- **Key update query($\mathcal{Q}_U$):** This query is useful for schemes with probabilistic attribute based private key generation, where a user of attribute set $\mathbb{S}$ may request for another attribute based private key after obtained the first copy. When $\mathfrak{A}$ makes a key update query for another attribute-based secret key after obtained the first copy. $\mathfrak{C}$ checks the list $\mathfrak{Q}$ for the tuple with the form $\langle \bar{j}, \mathbb{S}, SK_{\mathbb{S}} \rangle$. If there is no such tuple, then $\hat{j}$ is set to 1, and returns null. Otherwise, $\hat{j}$ is set to $\bar{j} + 1$, and returns $\hat{SK}_{\mathbb{S}} \leftarrow KeyGen(MSK, \mathbb{S})$. $\mathfrak{C}$ puts $\langle \hat{j}, \mathbb{S}, \hat{SK}_{\mathbb{S}} \rangle$ into the list $\mathfrak{Q}$, and returns the update times $\hat{j}$.

Challenge. $\mathfrak{A}$ outputs two messages $M_0, M_1 \in \mathcal{M}$ and a monotone access structure $\mathbb{A}^*$ such that $\forall \mathbb{S} \in \mathfrak{R}$ doesn't satisfy $\mathbb{A}^*$. $\mathfrak{C}$ randomly choose a bit $b \in \{0, 1\}$, and returns the cipher-text $CT^* \leftarrow Encrypt(M_b, \mathbb{A}^*)$.

Query 2. $\mathfrak{A}$ can make the key extraction queries like Query 1 except the queries on the attribute sets which satisfies $\mathbb{A}^*$.

Response. Finally, $\mathfrak{A}$ outputs a guess b' of b. $\mathfrak{A}$'s advantage of this game can be defined as $ADV_{\mathfrak{A}}(1^\lambda, \Sigma) = |2Pr[b = b'] - 1|$.

We say that a CP-ABE is AI-CPA secure w.r.t. auxiliary inputs from $\mathcal{F}$ on the condition that $ADV_{\mathfrak{A}}$ is negligible for any PPT adversary $\mathfrak{A}$ in the above game.

[4] $\bar{j}$ is used to index the attributes set and the secret key.

We consider the definition of function families $\mathcal{F}$. To parameterize $\mathcal{F}$, the min-entropy k_A[5] of attribute-based secret key is an important parameter. $\mathcal{F}$ can be denoted as $\mathcal{F}(g(k_A))$. Let q_l denote the times of $\mathfrak{A}$'s key leakage queries, and let q_e denote the times of $\mathfrak{A}$'s key extraction queries. Let Δ denote a set of q_e attribute-based secret keys. Then, for $\forall i \in [q_l]$, given $\{MPK, \mathbb{A}^*, \Delta, \{f_i(MSK, \mathfrak{Q}, MPK, \mathbb{S})\}_{i\in[q_l]}\}$, where all $f_i \in \mathcal{F}(g(k_A))$, no PPT algorithm can find a valid secret key $SK_{\mathbb{S}^*}$ such that attribute set $\mathbb{S}^* \in \mathbb{A}^*$ with a non-negligible probability greater than $g(k_A)$[6], where $g(k_A) \geq 2^{-k_A}$ is the hardness parameter. Our goal is to make $g(k_A)$ as close to $\text{negl}(k_A)$ as possible[7]. Thus, we have the following definition:

Definition 9 (AI-CPA-CP-ABE). *If a ciphertext-policy attribute-based encryption is CPA secure w.r.t. auxiliary input families $\mathcal{F}(g(k_A))$, then it is said to be $g(k_A)$ auxiliary input CPA secure ($g(k_A)$-AI-CPA).*

4 Construction of CP-ABE Resilient to Auxiliary Input Model

4.1 Preparation

Let Λ be a monotone universal attribute space. In the security proof of this construction, we should convert each attribute to a random number belonging to $\mathbb{Z}_N$, where N is a product of three distinct prime numbers p_1, p_2, p_3. Thus, an injection map χ should be pre-defined, and $\chi(S_i) \in \mathbb{Z}_N$ for all $S_i \in \Lambda$. Let $\Sigma = \chi(\Lambda)$, which is a subset of $\mathbb{Z}_N$. For simplicity, we denote the number set Σ as an universal attribute space in the following, and U denotes the cardinality of $U = |\Sigma|$.

Our construction should be resorted to the Goldreich-Levin Theorem for large fields. Let's review it according to [3,13].

Theorem 1 *(**GL Theorem for Large Fields**). Let q be a big prime, and let H be a subset of $GF(q)$. Let f mapping from H^m to $\{0,1\}^*$ be any function. Randomly chooses a vector s from H^m, and compute $y = f(s)$. Then, randomly selects a vector r from $GF(q)^m$. If a PPT distinguisher $\mathfrak{D}$ runs in time t, and there exists a probability ϵ such that*

$$|Pr[\mathfrak{D}(y, r, < r, s >) = 1] - Pr[u \leftarrow GF(q) : \mathfrak{D}(y, r, u) = 1]| = \epsilon,$$

then given $y \leftarrow f(s)$, there exists an inverter $\mathfrak{A}$ who can compute s from y in time $t' = t \cdot poly(m, |H|, 1/\epsilon)$ with the probability

$$Pr[s \leftarrow H^m, y \leftarrow f(s) : \mathfrak{A}(y) = s] \geq \frac{\epsilon^3}{512 \cdot m \cdot q^2}.$$

[5] If the key is generated randomly, then k_A equals the length of secret key.
[6] $g(k_A)$ is a non-negligible probability function.
[7] In the auxiliary model, any hard-to-invert function $f \in \mathcal{F}$ can hardly recover a secret key SK even the min-entropy of SK is 0.

4.2 Construction

Our construction is based on Waters' most efficient cipher-text-policy ABE scheme [10]. In our construction, we try to construct the public key as in the Yuen et al.'s scheme [3], then the master public key becomes $y_i = e(g_1, B_i)^{\alpha_i}$ (B_i, α_i are defined in the Setup algorithm.), and the master secret key becomes m pieces $(g_1^{\alpha_1}, \cdots, g_1^{\alpha_m})$ in order to use the GL Theorem for large fields.

Setup$(1^\lambda, \Sigma)$: The setup algorithm takes as input a security parameter λ, a monotone universal attribute space Σ. This algorithm runs the bilinear group generator to produce $\Theta = (N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, \hat{e})$, where p_1, p_2, p_3 are three distinct λ-bit primes. Then, it selects random generators $g_1, h_1, \cdots, h_U \in G_{p_1}$ and $g_3 \in G_{p_3}$. Let $m = (3\lambda)^{1/\epsilon}$, where the security is w.r.t. auxiliary inputs that are hard to invert with probability 2^{-m^ϵ}. It picks $a, \alpha_1, \cdots, \alpha_m, b_1, \cdots, b_m \in \mathbb{Z}_N$, and sets $B_1 = g_1^{b_1}, \cdots, B_m = g_1^{b_m}$. It selects $g_3 \in \mathbb{G}_{p_3}$ and $u_1, \cdots, u_m \in \mathbb{Z}_{p_3}$. The master public key is

$$MPK : \{\Theta, g_1, g_3, (g_1^{a/b_i})_{i\in[m]}, B_1, \cdots, B_m, h_1, \cdots, h_U, (y_i = e(g_1, B_i)^{\alpha_i})_{i\in[m]}\},$$

and the master secret key is $MSK = (g_1^{\alpha_i} \cdot g_3^{u_i})_{i\in[m]}$.

KeyGen$(MSK, MPK, \mathbb{S})$: This algorithm takes as input an attribute set $\mathbb{S}$[8], the master public key MPK and the master secret key MSK. It first chooses $y_{11}, \cdots, y_{1m}, y_2, y_{31}, \cdots, y_{3U}, t \in \mathbb{Z}_N$, and creates the private key as

$$\begin{aligned} SK_{\mathbb{S}} &= \{(K_{1i})_{i\in[m]}, K_2, (K_{3x})_{x\in\mathbb{S}}\} \\ &= \{(g_1^{\alpha_i} g_1^{at/b_i} \cdot g_3^{y_{1i}} g_3^{u_i})_{i\in[m]}, g_1^t g_3^{y_2}, (h_x^t g_3^{y_{3x}})_{x\in\mathbb{S}}\}. \end{aligned}$$

Encrypt(M, Π, MPK): The encryption algorithm takes as input an LSSS scheme $\Pi = (\mathcal{A}, \rho)$ for a monotone access structure $\mathbb{A}$, a message M and the master public key MPK. Here, $\mathcal{A}$ is an $l \times n$ matrix. The function ρ associates the rows of $\mathcal{A}$ to the attributes[9]. The algorithm first chooses random $s_1, \cdots, s_m \in \mathbb{Z}_N$ and a random vector $\boldsymbol{v} = (\sum_{i=1}^m s_i, v_2, \cdots, v_n) \in \mathbb{Z}_N^n$. For $i = 1$ to l, it computes $\lambda_i = \boldsymbol{v} \cdot \mathcal{A}_i$, where $\mathcal{A}_i$ is the vector corresponding to the ith row of $\mathcal{A}$. In addition, the algorithm chooses random $r_1, \cdots, r_l \in \mathbb{Z}_N$. The generated cipher-text CT is

$$\{C = M \cdot \prod_{i=1}^m y_i^{s_i}, (C_i' = B_i^{s_i})_{i\in[m]}, (C_i = g_1^{a\lambda_i} h_{\rho(i)}^{-r_i}, D_i = g_1^{r_i})_{i\in[l]}\}.$$

Decrypt(CT, SK, MPK): This algorithm takes as input a cipher-text CT for an LSSS scheme $\Pi = (\mathcal{A}, \rho)$ on the monotone access structure $\mathbb{A}$, a private key SK for a set $\mathbb{S}$ and the master public key MPK. If $\mathbb{S} \in \mathbb{A}$ is an authorized

[8] $\mathbb{S}$ is a subset of number set Σ.

[9] Since each attribute is mapped to a random number in $\mathbb{Z}_N$, ρ can be defined as $\rho : \mathbb{Z}_N^l \to \Sigma$.

set, then let $I \subset [l]$ be defined as $I = \{i : \rho(i) \in \mathbb{S}\}$. Then, it computes a set $\{\omega_i \in \mathbb{Z}_N\}_{i \in I}$ such that $\sum_{i \in I} \omega_i \lambda_i = \sum_{i=1}^{m} s_i$, if $\{\lambda_i\}$ are valid shares according to $\mathcal{A}$. Then, the decryption algorithm computes

$$\frac{\prod_{i=1}^{m} \hat{e}(C_i', K_{1i})}{\prod_{i \in I}(\hat{e}(C_i, K_2)\hat{e}(D_i, K_{3\rho(i)}))^{\omega_i}} = \prod_{i=1}^{m} y_i^{s_i}.$$

Finally, it can obtain the message M from C.

Correctness: The correctness of decryption is described as follows:

$$\begin{aligned}
&\frac{\prod_{i=1}^{m} \hat{e}(C_i', K_{1i})}{\prod_{i \in I}(\hat{e}(C_i, K_2)\hat{e}(D_i, K_{3\rho(i)}))^{\omega_i}} \\
&= \frac{\prod_{i=1}^{m} \hat{e}(B_i^{s_i}, g_1^{\alpha_i} \cdot g_1^{at/b_i} \cdot g_3^{y_{1i}+u_i})}{\prod_{i \in I}(\hat{e}(g_1^{a\lambda_i} h_{\rho(i)}^{-r_i}, g_1^t g_3^{y_2})\hat{e}(g_1^{r_i}, h_{\rho(i)}^t g_3^{y_{3\rho(i)}}))^{\omega_i}} \\
&= \frac{(\prod_{i=1}^{m} \hat{e}(g_1, B_i)^{\alpha_i s_i}) \cdot \hat{e}(g_1, g_1)^{at \cdot (\sum_{i=1}^{m} s_i)}}{\prod_{i \in I}(\hat{e}(g_1^{a\lambda_i}, g_1^t) \cdot \hat{e}(h_{\rho(i)}^{-r_i}, g_1^t) \cdot \hat{e}(g_1^{r_i}, h_{\rho(i)}^t))^{\omega_i}} \\
&= \frac{(\prod_{i=1}^{m} \hat{e}(g_1, B_i)^{\alpha_i s_i}) \cdot \hat{e}(g_1, g_1)^{at \cdot (\sum_{i=1}^{m} s_i)}}{\hat{e}(g_1, g_1)^{at \cdot \sum_{i \in I} \lambda_i \omega_i}} \\
&= \prod_{i=1}^{m} \hat{e}(g_1, B_i)^{\alpha_i s_i} = \prod_{i=1}^{m} y_i^{s_i}.
\end{aligned}$$

4.3 Performance Comparison

In this section, we provide the performance comparison with Lewko et al.'s scheme [27], Zhang et al.'s scheme [12] and our scheme. These three schemes are all ciphertext-policy attribute-based encryption schemes in the presence of key leakage model. Lewko et al.'s scheme and our scheme use LSSS to denote the access structure, while Zhang et al.'s scheme uses the minimal set to denote the access structure.

Let Pr denote the computation cost of pairing, Ex denote the exponent cost, and Mu denote the point multiplication. For [27] and our scheme, we assume that the LSSS matrix is $l \times n$. For [12,27], we assume that the leakage parameter is denoted as ϖ, the allowable leakage probability parameter is denoted as ς and the leakage bound of a key is denoted as ζ. For [12], let κ denote the number of minimal sets. In decryption, we only evaluate the computational costs of pairing, since the pairing operation is very time-consuming compared to other the other operations.

From the Table 1, we can see that the computational cost of Lewko et al.'s scheme [4] and Zhang et al.'s scheme [12] are mainly dependent on the leakage parameter ϖ, while the computational cost of our scheme is mainly dependent on the number of pieces m. However, Our scheme resilient to auxiliary input haven't the limitation of leakage bound.

Table 1. Performance comparison

Schemes	Lewko [27]	Zhang [12]	Our scheme
Encrypt	$2(\varpi+2l)Mu$	$(\varpi+2\kappa)Mu$	$(2l+m+1)Ex$
Decrypt	$(\varpi+2l+1)Pr$	$(\varpi+3)Pr$	$(m+2\|I\|)Pr$
Leakage bound	$\zeta=2+(\varpi-1-2\varsigma)\log p_2$	$\zeta=2+(\varpi-1-2\varsigma)\log p_2$	No
Leakage model	Bounded leakage	Continuous leakage	Auxiliary input

5 Security Proof

Our security proof employs the dual system encryption mechanism, which requires three semi-functional(SF) structures. Let g_2 be the generator of $\mathbb{G}_{p_2}$.

SF master secret key: $(g_1^{\alpha_i}\cdot g_2^{\theta_i}\cdot g_3^{u_i})_{i\in[m]}$, where $\theta_1,\cdots,\theta_m\in\mathbb{Z}_N$.

SF attribute-based secret key: $\{(g_1^{\alpha_i+at/b_i}\cdot g_2^{z_i}g_3^{y_{1i}})_{i\in[m]}, g_1^t g_2^d g_3^{y_2}, (h_x^t g_3^{y_{3x}})_{x\in\mathbb{S}}\}$, where $z_1,\cdots,z_m,d\in\mathbb{Z}_N$.

SF cipher-text: $\{C=M\cdot\prod_{i=1}^m y_i^{s_i}, (\tilde{C}_i'=B_i^{s_i}g_2^{\delta_i})_{i\in[m]}, (\tilde{C}_i=g_1^{a\lambda_i}h_{\rho(i)}^{-r_i}g_2^{\tau_i}, \tilde{D}_i=g_1^{r_i})_{i\in[l]}\}$, where $\delta_1,\cdots,\delta_m,\tau_1,\cdots,\tau_l\in\mathbb{Z}_N$.

When a SF attribute-based secret key is used to decrypt a SF cipher-text, we will obtain an extra term $\hat{e}(g_2,g_2)^{\sum_{i=1}^m\delta_i z_i-d\sum_{i\in I}\tau_i\omega_i}$. If $\sum_{i=1}^m\delta_i z_i-d\sum_{i\in I}\tau_i\omega_i=0$, we call a *SF attribute-based secret key* is a *nominally semi-functional(NSF) attribute-based secret key*. An NSF attribute-based secret key is a special kind of SF attribute-based secret key, which can be used to decrypt SF cipher-text, that means $\sum_{i=1}^m\delta_i z_i=d\sum_{i\in I}\tau_i\omega_i$. If an attribute-based secret key is generated from a SF master secret key, then it is also semi-functional. If we use it to decrypt a SF cipher-text, we will obtain another extra term $\hat{e}(g_2,g_2)^{\sum_{i=1}^m\delta_i\theta_i-d\sum_{i\in I}\tau_i\omega_i}$. Similarly, if $\sum_{i=1}^m\delta_i\theta_i=d\sum_{i\in I}\tau_i\omega_i$, then decryption still works and the SF attribute-based secret key is an NSF attribute-based secret key.

Theorem 2. *Our CP-ABE scheme is (2^{-m^ϵ})-AI-CPA leakage secure under the modified assumptions 1,2 and 3.*

Proof: We prove this theorem by a series of games. In the first real $Game_{rl}$, the key and cipher-text are normal forms. Let $\mathbb{A}^*$ denote the challenge access structure, which is a monotone collection. The second game $Game_{rt}$ is the same as $Game_{rl}$ except that the adversary cannot ask for any attribute set belonging to the collection $\mathbb{A}^*$. Then, we convert the challenge cipher-text into semi-functional, and convert the keys into semi-functional forms one by one. Finally, we also prove that the message is distinguishable from a random message in the challenge cipher-text.

Lemma 1. *If $Adv_A^{Game_{rl}}-Adv_A^{Game_{rt}}\geq\epsilon$, then Assumption 2 is broken.*

Lemma 2. *If $Adv_A^{Game_{rt}}-Adv_A^{Game_0}\geq\epsilon$, then modified Assumption 1 is broken.*

Lemma 3. *If* $Adv_A^{Game_{k+1}} - Adv_A^{Game_k} \geq \epsilon$, *then modified Assumption 2 is broken.*

Lemma 4. *If* $Adv_A^{Game_Q} - Adv_A^{Game_f} \geq \epsilon$, *then modified Assumption 3(modified BSDP assumption) is broken.*

We prove **Lemma 1–4** in **Appendix** B. From **Lemma 1–4**, if modified assumptions 1,2,3 hold, then $Game_{rl}$ is indistinguishable from $Game_f$. Obviously, the adversary can win the $Game_{rl}$ with negligible probability. Thus, our CP-ABE scheme is (2^{-m^ϵ})-AI-CPA leakage secure. □

Note: Our scheme can be easily extended to an ABE scheme secure in the *continual auxiliary leakage* model [3], if the extended scheme does not allow leakage during the setup phase. It only adds two update algorithms for MSK and attribute-based secret key, and the proof is similar to the above, since the updates all used random elements in $\mathbb{G}_{p3}$, which has no impact to the previous proof.

6 KP-ABE Resilient to Auxiliary Input

In this section, we construct key-policy attribute-based encryption leakage resilient to auxiliary input model. In KP-ABE, a key is associated with an access structure and a cipher-text is associated with a set of attributes. The construction has the similar security proof with AI-CP-ABE. In this construction, we encode a monotone universal attribute space as an index numbers set, which is still monotone. Let Σ be the universal attribute space, and $U = |\Sigma|$. We use a function I to map each attribute to its index number. Let $I_{\mathbb{S}}$ denote the index numbers set of attributes set $\mathbb{S}$. It means that $I_{\mathbb{S}} \subset \{1, \cdots, U\}$. Let $\mathbb{A} = \{\mathbb{S}_1, \cdots, \mathbb{S}_n\}$ be a monotone access structure, and all $\mathbb{S}_i$s are authorized attribute sets. Then, $I_{\mathbb{A}} = \{I_{\mathbb{S}_1}, \cdots, I_{\mathbb{S}_n}\}$ is a monotone collection of index numbers sets corresponding to $\mathbb{A}$.

Setup$(1^\lambda, \Sigma)$: The setup algorithm takes as input a security parameter λ, a monotone universal attributes set Σ. This algorithm runs the bilinear group generator to produce $\Theta = (N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, \hat{e})$. Then, it selects random generators $g_1, h_1, \cdots, h_u \in G_{p_1}$ and $g_3 \in G_{p_3}$. Let $m = (3\lambda)^{1/\epsilon}$. It picks $a, \alpha_1, \cdots, \alpha_m\, b_1, \cdots, b_m \in \mathbb{Z}_{p_1}$, and sets $B_1 = g_1^{b_1}, \cdots, B_m = g_1^{b_m}$. It selects $g_3 \in \mathbb{G}_{p_3}$ and $u \in \mathbb{Z}_{p_3}$. The master public key is

$$MPK : \{\Theta, g_1, g_1^a, g_3, B_1, \cdots, B_m, h_1, \cdots, h_U, (H_i = h_i^{b_1}, \cdots, H_i = h_i^{b_m})_{i\in[U]},$$
$$(y_i = e(g_1, B_i)^{\alpha_i})_{i\in[m]}\},$$

and the master secret key is $MSK = (g_1^{\alpha_i} g_3^{u_i})_{i\in[m]}$.

KeyGen$(MSK, \mathbb{A})$: This algorithm takes as input a monotone access structure $\mathbb{A}$ and the master secret key MSK. It first chooses

$$y_{11}, \cdots, y_{1m}, y_2, y_{31}, \cdots, y_{3U}, t, r_1, \cdots, r_U \in \mathbb{Z}_N,$$

and creates the private key as

$$SK = \{(K_{1i})_{i\in[m]}, (K_{2i})_{i\in I_{\mathbb{A}}}, (K_{3i})_{i\in[U]}\}$$
$$= \{(g_1^{\alpha_i+at} \cdot g_3^{y_{1i}+u_i})_{i\in[m]}, (g_1^{-at}(\prod_{j\in I_{\mathbb{S}_i}} h_j^{r_j})g_3^{y_2})_{i\in I_{\mathbb{A}}}, (g_1^{r_i} g_3^{y_{3i}})_{i\in[U]}\}.$$

Encrypt$(M, \mathbb{S})$: The encryption algorithm takes as input an attributes set $\mathbb{S}$ and a message M. The algorithm firstly transforms $\mathbb{S}$ to its corresponding index set $I_{\mathbb{S}}$. Then, it chooses random $s_1, \cdots, s_m \in \mathbb{Z}_N$, and outputs the cipher-text CT as

$$< C = M \cdot \prod_{i=1}^{m} y_i^{s_i}, (C_i' = B_i^{s_i})_{i\in[m]}, (C_i = \prod_{j=1}^{m} H_{ij}^{s_j})_{i\in I_{\mathbb{S}}} > .$$

Decrypt(CT, SK): This algorithm takes as input a cipher-text CT for an attribute set $\mathbb{S}_k$ and a private key SK associated with a monotone access structure $\mathbb{A}$. If $\mathbb{S}_k \in \mathbb{A}$ is an authorized set, then $I_{\mathbb{S}_k} \in I_{\mathbb{A}}$. The decryption algorithm computes

$$\frac{\prod_{i=1}^{m} \hat{e}(C_i', K_{1i}) \cdot \hat{e}(K_{2k}, \prod_{i=1}^{m} C_i')}{\prod_{i\in I_{\mathbb{S}_k}} \hat{e}(C_i, K_{3i})} = \prod_{i=1}^{m} y_i^{s_i}.$$

Finally, it can obtain the message M from C.

7 Conclusions

In this paper, we propose a security model of CP-ABE leakage resilient to auxiliary input, and a concrete construction based on linear secret sharing schemes. Our scheme can tolerate leakage on master key and attribute-based secret key with auxiliary input. For the security proof of our scheme, we present three modified static assumptions in composite order bilinear groups, and prove them in detail. Our scheme also can be easily extended to an ABE scheme resilient to continual auxiliary leakage, if it doesn't allow leakage in setup phase. Finally, we also propose a KP-ABE scheme resilient to auxiliary input.

Acknowledgments. This research is partially supported by the National Natural Science Foundation of China under Grant No.61373006, NSFC/RGC Joint Research Scheme of Hong Kong and China (N-HKU 729/13) and seed funding projects of HKU (201311159040 and 201411159142).

A Proofs of Three Modified Assumptions

We adopt the notion of [4] to denote an element $g_1^{a_1} g_2^{a_2} g_3^{a_3}$ of $\mathbb{G}$ as (a_1, a_2, a_3). The element $\hat{e}(g_1, g_1)^{a_1} \hat{e}(g_2, g_2)^{a_2} \hat{e}(g_3, g_3)^{a_3}$ in $\mathbb{G}_T$ will be denoted by $[a_1, a_2, a_3]$.

We use capital letter to denote the random variables. For example, $X = (X_1, Y_1, Z_1)$ is denoted as a random element of $\mathbb{G}$. We say that X is dependent on $\{A_i\}$, if there exists values $\lambda_i \in \mathbb{Z}_N$ such that $X = \sum_i \lambda_i A_i$. Otherwise, X is independent on $\{A_i\}$. For the security proof, we should review the following two theorems from [9].

Theorem 3 *(Theorem A.1 in [9]). Let $N = \prod_{i=1}^{m} p_i$ be a product of distinct primes, each greater than 2^λ. Let $\{A_i\}$ be a random variables set over $\mathbb{G}$, and let $\{B_i\}, T_0, T_1$ be random variables over $\mathbb{G}_T$, where all variables have the degree greater than t. The following game between an adversary $\mathfrak{A}$ and a challenger $\mathfrak{C}$ is in generic group model.*

Given $N, \{A_i\}, \{B_i\}$, $\mathfrak{C}$ chooses a random bit b, and sends T_b to $\mathfrak{A}$. $\mathfrak{A}$ outputs a bit b', and succeeds the game if $b' = b$.

If the following conditions are satisfied, then $\mathfrak{C}$ can find a nontrivial factor of N by using $\mathfrak{A}$ in time polynomial in λ with probability at least $\delta - \mathcal{O}(q^2 t/2^\lambda)$.

1. *Each of T_0 and T_1 is independent of $\{B_i\} \cup \{e(A_i, A_j)\}$.*
2. *$\mathcal{A}$ issuing at most q queries and having advantage δ in the above game.*

Theorem 4 *(Theorem A.2 in [9]). Let $N = \prod_{i=1}^{m} p_i$ be a product of distinct primes, each greater than 2^λ. Let $\{A_i\}$ be a random variables set over $\mathbb{G}$, and let $\{B_i\}, T_0, T_1$ be random variables over $\mathbb{G}_T$, where all variables have the degree greater than t. The game between an adversary $\mathfrak{A}$ and a challenger $\mathfrak{C}$ is the same as above.*

Let $S := \{i | \hat{e}(T_0, A_i) \neq \hat{e}(T_1, A_i)\}$. If the following conditions are satisfied, then $\mathfrak{C}$ can find a nontrivial factor of N by using $\mathfrak{A}$ in time polynomial in λ with probability at least $\delta - \mathcal{O}(q^2 t/2^\lambda)$.

1. *Each of T_0 and T_1 is independent of $\{A_i\}$.*
2. *For all $k \in S$, $\hat{e}(T_0, A_k)$ and $\hat{e}(T_1, A_k)$ are independent of $\{B_i\} \cup \{\hat{e}(A_i, A_j)\} \cup \{\hat{e}(T_1, A_i)\}_{i \neq k}$.*
3. *$\mathcal{A}$ issuing at most q queries and having advantage δ in the above game.*

We apply these two theorems to prove the hardness of our modified assumptions in generic group model.

modified 1-SDP assumption. To prove this assumption, we will use **Theorem 4**. Firstly, we can express this assumption as:

$$A_1 = (1, 0, 0), A_2 = (0, 0, X_3)$$
$$\{T_{0i} = (X_{1i}, X_{2i}, 0)\}_{i \in [m]}, \{T_{1i} = (X_{1i}, 0, 0)\}_{i \in [m]}$$

Since $\hat{e}(T_{0i}, A_1) = [X_{1i}, 0, 0] = \hat{e}(T_{1i}, A_1) = [X_{1i}, 0, 0]$ and $\hat{e}(T_{0i}, A_2) = [0, 0, 0] = \hat{e}(T_{1i}, A_2) = [0, 0, 0]$, we can note that $S = \emptyset$, and for all $i \in [m]$, T_{0i} and T_{1i} are independent of $\{A_1, A_2\}$ since X_{1i} does not exist in both A_1 and A_2. Then, in the game of Theorem 4, if $\exists i \in [m]$, the adversary $\mathfrak{A}$ can distinguish T_{0i} and T_{1i} with probability δ, then N can be factored with probability less than δ. Since it

is hard to find a nontrivial factor of N, then the modified 1-SDP assumption is secure.

modified 2-SDP assumption. To prove this assumption, we will also use **Theorem 4**. Firstly, we can express this assumption as:

$$A_1 = (1,0,0), \{A_{2i} = (X_{1i}, X_{2i}, 0)\}_{i\in[m]}, A_3 = (0,0,X_3), A_4 = (0, Y_2, Y_3)$$

$$T_0 = (Z_1, Z_2, Z_3), T_1 = (Z_1, 0, Z_3)$$

We note that $S = \{\{2i\}_{i\in[m]}, 4\}$ in this case. It is clear that

1. Both T_0 and T_1 are independent of $\{A_i\}$, since Z_1 cannot be found in A_i's.
2. Since $\hat{e}(T_0, A_{2i}) = [X_{1i}Z_1, X_{2i}Z_2, 0]$,

$$\{\hat{e}(T_0, A_i)\}_{i\in\{1,3,4\}} = \{[Z_1, 0, 0], [0, 0, X_3Z_3], [0, Y_2Z_2, Y_3Z_3]\}$$

and

$$\{\hat{e}(T_0, A_{2j})\}_{j\in[m], j\neq i} = \{X_{1j}Z_1, X_{2j}Z_2, 0\}_{j\in[m], j\neq i}$$

$\hat{e}(T_0, A_{2i})$ is independent of $\{\hat{e}(A_i, A_j)\} \cup \{\hat{e} = (T_0, A_i)\}_{i\in\{1,3,4\}} \cup \{\hat{e} = (T_0, A_{2j})\}_{j\in[m],j\neq i}$. We can find that it is impossible to obtain $X_{1i}Z_1$ in the first coordinate of a combination of elements of $\{\hat{e}(A_i, A_j)\} \cup \{\hat{e} = (T_0, A_i)\}_{i\in\{1,3,4\}} \cup \{\hat{e} = (T_0, A_{2j})\}_{j\in[m],j\neq i}$. Obviously, $\hat{e}(T_1, A_{2i})$ is also independent of $\{\hat{e}(A_i, A_j)\} \cup \{\hat{e} = (T_0, A_i)\}_{i\in\{1,3,4\}} \cup \{\hat{e} = (T_0, A_{2j})\}_{j\in[m],j\neq i}$ due to the same reason.
3. From $\hat{e}(T_0, A_4) = [0, Y_2Z_2, Y_3Z_3]$ and $\hat{e}(T_1, A_4) = [0, 0, Y_3Z_3]$, we can conclude that $\hat{e}(T_0, A_4)$ and $\hat{e}(T_1, A_4)$ are both independent of $\{\hat{e}(A_i, A_j)\} \cup \{\hat{e} = (T_0, A_i)\}_{i\neq 4}$, since we cannot obtain Y_3Z_3 in the third coordinate of a combination of elements of $\{\hat{e}(A_i, A_j)\} \cup \{\hat{e} = (T_0, A_i)\}_{i\neq 4}$.

Thus, from Theorem 4, modified 2-SDP assumption is generically secure on the condition that it is hard to factor N.

modified BSDP assumption. We use **Theorem** 3. to prove this assumption. Firstly, we can express this assumption as:

$$A_1 = (1,0,0), \{A_{2i} = (1/b_i, 0, 0)\}_{i\in[m]}, \{A_{3i} = (b_i\alpha_i, X_2, 0)\}_{i\in[m]}, A_4 = (0,0,X_3),$$

$$\{A_{5i} = (b_i s_i, Y_2, 0)\}_{i\in[m]}, A_6 = (0, Z_2, 0), T_0 = [\sum_{i=1}^{m} \alpha_i b_i s_i, 0, 0], T_1 = [Z_1, Z_2, Z_3].$$

We note that

1. It is clear that the only way to obtain $\sum_{i=1}^{m} \alpha_i b_i s_i$ is to compute $\prod_{i=1}^{m} \hat{e}(A_{3i}, A_{5i})$. However, $\prod_{i=1}^{m} \hat{e}(A_{3i}, A_{5i}) = [\sum_{i=1}^{m} \alpha_i b_i s_i, (X_2Y_2)^m, 0]$, then $(X_2Y_2)^m$ are left in the second coordinate that cannot be canceled. So T_0 is independent of $\{\hat{e}(A_i, A_j)\}$.
2. T_1 is independent of $\{\hat{e}(A_i, A_j)\}$, because Z_1, Z_2, Z_3 cannot be found in $\{A_i\}$.

From the discussion above, we can conclude that the modified BSDP assumption is generically secure under Theorem 3.

B Proofs of Lemma 1–4

Lemma 1. *If* $Adv_A^{Game_{rl}} - Adv_A^{Game_{rt}} \geq \epsilon$, *then Assumption 2 is broken.*

Proof: Let $\mathbb{A}^*$ denote the challenge access structure. For every $\mathcal{S}^* \in \mathbb{A}^*$, assuming that $\mathcal{S}^* = \{S_1, \cdots, S_n\}$ has n attributes[10], we define a superset of $\mathcal{S}^*$ as $\mathbb{S}^* = \{S_1' | S_1' = S_1 \mod p_2\} \cup \cdots \cup \{S_n' | S_n' = S_n \mod p_2\}$. Let Ω^* denote the collection of all $\mathbb{S}^*$s. If adversary $\mathfrak{A}$ makes key query on an attribute set $\Xi \notin \mathbb{A}^*$, for $\forall S_i' \in \Xi$, the challenger $\mathfrak{C}$ answers as follows:

- If $S_i' \notin \mathbb{S}^*$, for $\forall \mathbb{S}^* \in \Omega^*$, then $\mathfrak{C}$ responses by using MSK and the **KeyGen** algorithm.
- If $S_i' \in \mathbb{S}^*$, for $\exists \mathbb{S}^* \in \Omega^*$, then $S_i' \neq S_i$ and $S_i' = S_i \mod p_2$. $\mathfrak{C}$ computes $a = \gcd(S_i - S_i^*, N)$. We denote $b = N/a$, where $N = p_1 p_2 p_3$. We assume that $(g, X_1X_2, X_3, Y_2Y_3, T)$ is an instance from 2-SDP assumption.
  1. If $a = p_1p_2$ and $b = p_3$, then $\mathfrak{C}$ can check whether $a = p_1p_2$ from $(X_1X_2)^a = 1$. If the equation holds, then $\mathfrak{C}$ can distinguish between $T \in \mathbb{G}_{p_1p_3}$ and $T \in \mathbb{G}$ by using $\hat{e}(Y_2Y_3, T)^b \stackrel{?}{=} 1$.
  2. If $a = p_2p_3$ and $b = p_1$, then $\mathfrak{C}$ checks whether $a = p_2p_3$ from $(Y_2Y_3)^a = 1$. $\mathfrak{C}$ also can distinguish between $T \in \mathbb{G}_{p_1p_3}$ and $T \in \mathbb{G}$ by using $\hat{e}(X_1X_2, T)^b \stackrel{?}{=} 1$.
  3. If $a = p_2$ and $b = p_1p_3$, then $\mathfrak{C}$ can distinguish between $T \in \mathbb{G}_{p_1p_3}$ and $T \in \mathbb{G}$ by using $T^b \stackrel{?}{=} 1$. □

Then, the challenge ciphertext is converted into semi-functional in $Game_0$.

Lemma 2. *If* $Adv_A^{Game_{rt}} - Adv_A^{Game_0} \geq \epsilon$, *then modified Assumption 1 is broken.*

Proof: Given an instance $(N, g_1, X_3, \mathbb{G}, \mathbb{G}_T, (T_i)_{i\in[m]})$ of modified 1-SDP assumption, $\mathfrak{C}$ constructs the master public key MPK as

$$< g_1, X_3, (g_1^{a/b_i})_{i\in[m]}, B_1, \cdots, B_m, h_1, \cdots, h_U, (y_i = \hat{e}(g_1, B_i)^{\alpha_i})_{i\in[m]} >,$$

where $a, \alpha_i, b_i \in \mathbb{Z}_N$. The master secret key $MSK = (g_1^{\alpha_i} X_3^{u_i})_{i\in[m]}$. $\mathfrak{C}$ can answer the key extraction queries, key leakage queries and key update queries from $\mathfrak{A}$. In the challenge phase, $\mathfrak{A}$ provides the challenge message and access structure as $(M_0, M_1, \mathbb{A}^*)$. Then, $\mathfrak{C}$ randomly chooses values $\tilde{\lambda}_1, \cdots, \tilde{\lambda}_l, r_1, \cdots, r_l \in \mathbb{Z}_N$, and outputs the ciphertext CT^* as

$$< M_b \cdot \prod_{i=1}^{m} \hat{e}(g_1^{\alpha_i}, T_i), (T_i)_{i\in[m]}, (C_i = T_i^{a\tilde{\lambda}_i} h_{\rho(i)}^{-r_i}, D_i = g_1^{r_i})_{i\in[l]} >$$

If $T_i = g_1^{b_i s_i} g_2^{c_i} \in \mathbb{G}_{p_1p_2}$, then CT^* is

$$< M_b \cdot \prod_{i=1}^{m} \hat{e}(g_1^{\alpha_i}, B_i^{s_i}), (B_i^{s_i} g_2^{\delta_i})_{i\in[m]}, (C_i = g_1^{a\lambda_i} h_{\rho(i)}^{-r_i} g_2^{\tau_i}, D_i = g_1^{r_i})_{i\in[l]} >,$$

[10] Here, each attribute is mapped to a random number in $\mathbb{Z}_N$.

where $\delta_i = c_i, \lambda_i = b_i \cdot s_i \cdot \tilde{\lambda}_i, \tau_i = ac\tilde{\lambda}_i$. This is a semi-functional ciphertext, and $\mathfrak{C}$ simulates $Game_0$. If $T_i \in \mathbb{G}_{p_1}$, $\mathfrak{C}$ can simulate a normal ciphertext game $Game_{rt}$. Thus, if $\mathfrak{A}$ can distinguish between a semi-functional ciphertext and a normal ciphertext with a non-negligible probability, then $\mathfrak{C}$ can use $\mathfrak{A}$'s output to break the modified Assumption 1. □

Let Q denote the times of queries that $\mathfrak{A}$ issues when the challenge ciphertext is semi-functional. We set two types of attribute-based private key as follows:

Type I: $< (g_1^{\alpha_i + at/b_i} \cdot g_2^{z_i} g_3^{y_{1i}+u_i})_{i\in[m]}, g_1^t g_2^d g_3^{y_2}, (h_x^t g_3^{y_{3x}})_{x\in\mathbb{S}} >$
Type II: $< (g_1^{\alpha_i + at/b_i} \cdot g_3^{y_{1i}+u_i})_{i\in[m]}, g_1^t g_2^d g_3^{y_2}, (h_x^t g_3^{y_{3x}})_{x\in\mathbb{S}} >$

For $k = 1, \cdots, Q-1$, in $Game_k$, the first $k-1$ keys are semi-functional of type II, the k-th key is semi-functional of type I, and the rest keys are normal. Thus, in $Game_Q$, all keys are semi-functional of type II.

Lemma 3. *If $Adv_A^{Game_{k+1}} - Adv_A^{Game_k} \geq \epsilon$, then modified Assumption 2 is broken.*

Proof: Provided an instance $(g_1, (X_{1i}X_{2i})_{i\in[m]}, X_3, Y_2Y_3, T)$ of modified 2-SDP assumption, $\mathfrak{C}$ constructs the master public key

$$MPK :< \Theta, g_1, g_3, (g_1^{a/b_i})_{i\in[m]}, B_1, \cdots, B_m, h_1, \cdots, h_U, (y_i = e(g_1, B_i)^{\alpha_i})_{i\in[m]} >,$$

and the master secret key $MSK = (g_1^{\alpha_i} g_3^{u_i})_{i\in[m]}$. In the first $k-1$ key queries, $\mathfrak{C}$ answers with $< (g_1^{\alpha_i + at/b_i} \cdot g_3^{y_{1i}+u_i})_{i\in[m]}, g_1^t (Y_2Y_3)^h g_3^{y_2}, (h_x^t g_3^{y_{3x}})_{x\in\mathbb{S}} >$, which is a type II semi-functional key. For $k+1$-th to Q-th queries, $\mathfrak{C}$ answers with normal keys.

For the k-th query, $\mathfrak{C}$ answers the key as follows:

1. $< (g_1^{\alpha_i} \cdot T^a \cdot g_3^{y_{1i}+u_i})_{i\in[m]}, T \cdot g_3^{y_2}, (h_x^t g_3^{y_{3x}})_{x\in\mathbb{S}} >$
2. $< (g_1^{\alpha_i} \cdot T^a \cdot g_3^{y_{1i}+u_i})_{i\in[m]}, T \cdot g_3^{y_2} \cdot (Y_2Y_3)^h, (h_x^t g_3^{y_{3x}})_{x\in\mathbb{S}} >$

In case 1, if $T = g_1^t g_2^r g_3^s \in \mathbb{G}$, then the k-th key is a semi-functional key of type I. If $T = g_1^t g_3^s \in \mathbb{G}_{p_1p_3}$, the k-th key is a normal form key.

In case 2, if $T = g_1^t g_2^r g_3^s \in \mathbb{G}$, then the k-th key is a semi-functional key of type I. However, if $T = g_1^t g_3^s \in \mathbb{G}_{p_1p_3}$, the k-th key is a type II semi-functional key.

When $\mathfrak{A}$ makes a key leakage query, $\mathfrak{C}$ returns $f(MSK', \mathfrak{Q}, MPK, \mathbb{S})$, where MSK' is semi-functional, and for the last entry $< \cdot, \mathbb{S}, SK'_{\mathbb{S}} > \in \mathfrak{Q}$, $SK'_{\mathbb{S}}$ is a type II semi-functional key.

When $\mathfrak{A}$ makes a key update query, $\mathfrak{C}$ returns a type II semi-functional key $SK'_{\mathbb{S}}$ and the update times j', then puts $< j', \mathbb{S}, SK'_{\mathbb{S}} >$ to $\mathfrak{Q}$.

In the challenge phase, $\mathfrak{C}$ randomly chooses $\tilde{\lambda}_1, \cdots, \tilde{\lambda}_l \in \mathbb{Z}_N$, and returns the ciphertext as

$$C = M_b \prod_{i=1}^{m} \hat{e}(g_1^{\alpha_i}, X_{1i}X_{2i}), (C'_i = X_{1i}X_{2i})_{i\in[m]}, (C_i = (X_{1i}X_{2i})^{a\tilde{\lambda}_i} h_{\rho(i)}^{-r_i}, D_i = g_1^{r_i})_{i\in[l]}.$$

If we let $X_{1i}X_{2i} = g_1^{b_i s_i} g_2^{c_i}$, then

$$C = M_b \prod_{i=1}^{m} \hat{e}(g_1^{\alpha_i}, B_i^{s_i}), (C_i' = B_i^{s_i} g_2^{\delta_i})_{i\in[m]}, (C_i = g_1^{a\lambda_i} h_{\rho(i)}^{-r_i} g_2^{\tau_i}, D_i = g_1^{r_i})_{i\in[l]},$$

where $\delta_i = c_i, \lambda_i = b_i \cdot s_i \cdot \tilde{\lambda}_i, \tau_i = ac\tilde{\lambda}_i$. This is a semi-functional ciphertext.

We can thus conclude that, if $T \in \mathbb{G}$, $\mathfrak{C}$ can simulate $Game_{k+1}$. Otherwise, $\mathfrak{C}$ can simulate $Game_k$. From the above analysis, $\mathfrak{A}$ cannot distinguish between type I semi-functional key and normal form key in case 1, and $\mathfrak{A}$ also cannot distinguish between type I semi-functional key and type II semi-functional key in case 2. Thus, if an adversary has a non-negligible probability in $Adv_A^{Game_{k+1}} - Adv_A^{Game_k}$, then $\mathfrak{C}$ can break the modified 2-SDP assumption. □

The final game $Game_f$ is the same as $Game_Q$ except that the message is masked with a random element in $\mathbb{G}_T$, instead of M_0, M_1. That is to say, the value of b is information theoretically hidden from $\mathfrak{A}$.

Lemma 4. *If $Adv_A^{Game_Q} - Adv_A^{Game_f} \geq \epsilon$, then modified Assumption 3(modified BSDP assumption) is broken.*

Proof: Given an instance $(g_1, (g_1^{1/b_i})_{i\in[m]}, (B_i^{\alpha_i} X_2)_{i\in[m]}, X_3, (B_i^{s_i} Y_2)_{i\in[m]}, Z_2, T)$ of modified BSDP assumption, $\mathfrak{C}$ sets $g_3 = X_3, g_2 = Z_2, y_i = \hat{e}(g_1, B_i^{\alpha_i} X_2) = \hat{e}(g_1, B_i)^{\alpha_i}$. $\mathfrak{C}$ constructs the master public key MPK and the master secret key $MSK = (B_i^{\alpha_i} X_2 \cdot g_3^{u_i})_{i\in[m]}$.

In key extraction phase, $\mathfrak{C}$ can answer all key queries as

$$\begin{aligned} SK_{\mathbb{S}} &= < (K_{1i})_{i\in[m]}, K_2, (K_{3x})_{x\in\mathbb{S}} > \\ &= < ((B_i^{\alpha_i} X_2) \cdot g_1^{at/b_i} \cdot g_3^{y_{1i}+u_i})_{i\in[m]}, g_1^t g_3^{y_2}, (h_x^t g_3^{y_{3x}})_{x\in\mathbb{S}} > . \end{aligned}$$

$\mathfrak{C}$ also can answer the key leakage queries and key update queries from $\mathfrak{A}$, since it knows MSK.

In the challenge phase, $\mathfrak{C}$ randomly chooses $\tilde{\lambda}_1, \cdots, \tilde{\lambda}_l, r_1, \cdots, r_l \in \mathbb{Z}_N$ returns the ciphertext CT^* as

$$< M_b \cdot T, (B_i^{s_i} Y_2)_{i\in[m]}, (C_i = (B_i^{s_i} Y_2)^{a\tilde{\lambda}_i} h_{\rho(i)}^{-r_i}, D_i = g_1^{r_i})_{i\in[l]} >,$$

where T is the assumption term. Let $B_i^{s_i} Y_2 = B_i^{s_i} g_2^{c_i}$, then

$$< M_b \cdot T, (B_i^{s_i} g_2^{\delta_i})_{i\in[m]}, (C_i = g_1^{a\lambda_i} h_{\rho(i)}^{-r_i} g_2^{\tau_i}, D_i = g_1^{r_i})_{i\in[l]} >,$$

where $\delta_i = c_i, \lambda_i = b_i \cdot s_i \cdot \tilde{\lambda}_i, \tau_i = ac\tilde{\lambda}_i$. If $T = \prod_{i=1}^{m} \hat{e}(g_1, B_i)^{\alpha_i s_i}$, then CT^* is a semi-functional ciphertext and $\mathfrak{C}$ can simulate $Game_Q$ in this case. However, if $T \in \mathbb{G}_T$ is random element, then $\mathfrak{C}$ can simulate $Game_f$. Thus, if the adversary $\mathfrak{A}$ has non-negligible for distinguishing between $Game_f$ and $Game_Q$, then $\mathfrak{C}$ can break the modified BSDP assumption by using $\mathfrak{A}$'s output with the same probability. □

References

1. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
2. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
3. Yuen, T.H., Chow, S.S.M., Zhang, Y., Yiu, S.M.: Identity-based encryption resilient to continual auxiliary leakage. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 117–134. Springer, Heidelberg (2012)
4. Lewko, A., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455–479. Springer, Heidelberg (2010)
5. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
6. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
7. Beimel, A., Gal, A., Paterson, M.: Lower bounds for monotone span programs. Comput. Complex. **6**(1), 29–45 (1997)
8. Pandit, T., Barua, R.: Efficient fully secure attribute-based encryption schemes for general access structures. In: Takagi, T., Wang, G., Qin, Z., Jiang, S., Yu, Y. (eds.) ProvSec 2012. LNCS, vol. 7496, pp. 193–214. Springer, Heidelberg (2012)
9. Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009)
10. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
11. Beimel, A.: Secure schemes for secret sharing and key distribution. Ph.D. thesis, Israel Institute of Technology, Technion, Haifa, Israel (1996)
12. Zhang, M., Shi, W., Wang, C., Chen, Z., Mu, Y.: Leakage-resilient attribute-based encryption with fast decryption: models, analysis and constructions. In: Deng, R.H., Feng, T. (eds.) ISPEC 2013. LNCS, vol. 7863, pp. 75–90. Springer, Heidelberg (2013)
13. Dodis, Y., Goldwasser, S., Tauman Kalai, Y., Peikert, C., Vaikuntanathan, V.: Public-key encryption schemes with auxiliary inputs. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 361–381. Springer, Heidelberg (2010)
14. Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (2009)
15. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 113–134. Springer, Heidelberg (2010)
16. Dodis, Y., Lewko, A., Waters, B., Wichs, D.: Storing secrets on continually leaky devices. In: FOCS 2011, pp. 688–697 (2011)

17. Yang, B., Zhang, M.: LR-UESDE: a continual-leakage resilient encryption with unbounded extensible set delegation. In: Takagi, T., Wang, G., Qin, Z., Jiang, S., Yu, Y. (eds.) ProvSec 2012. LNCS, vol. 7496, pp. 125–142. Springer, Heidelberg (2012)
18. Zhang, M., Yang, B., Takagi, T.: Bounded leakage-resilient funtional encryption with hidden vector predicate. Comput. J. **56**(4), 464–477 (2013). Oxford
19. Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., Sahai, A.: Exposure-resilient functions and all-ornothing transforms. In: EUROCRYPT, pp. 453–469 (2000)
20. Kamp, J., Zuckerman, D.: Deterministic extractors for bit- xing sources and exposure-resilient cryptography. In: FOCS, pp. 92–101 (2003)
21. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS, pp. 293–302 (2008)
22. Faust, S., Kiltz, E., Pietrzak, K., Rothblum, G.N.: Leakage-resilient signatures. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 343–360. Springer, Heidelberg (2010)
23. Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., Wichs, D.: Public-key encryption in the bounded-retrieval model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 113–134. Springer, Heidelberg (2010)
24. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer, Heidelberg (2009)
25. Di Crescenzo, G., Lipton, R.J., Walfish, S.: Perfectly secure password protocols in the bounded retrieval model. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 225–244. Springer, Heidelberg (2006)
26. Dodis, Y., Haralambiev, K., Lopez-Alt, A., Wichs, D.: Cryptography against continuous memory attacks. In: FOCS, pp. 511–520 (2010)
27. Lewko, A., Rouselakis, Y., Waters, B.: Achieving leakage resilience through dual system encryption. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 70–88. Springer, Heidelberg (2011)