A^{2}BE: Accountable Attribute-Based Encryption for Abuse Free Access Control

Jin Li[1], Kui Ren[1], and Kwangjo Kim[2]

[1] Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, Illinois 60616, USA
{jin.li,kren@ece.iit.edu}

[2] International Research center for Information Security (IRIS)
Information and Communications University(ICU)
103-6 Munji-Dong, Yuseong-Gu, Daejeon, 305-732, Korea
{kkj@icu.ac.kr}

Abstract. As a recently proposed public key primitive, attribute-based encryption (ABE) (including Ciphertext-policy ABE (CP-ABE) and Key-policy ABE (KP-ABE)) is a highly promising tool for secure access control. In this paper, the issue of key abuse in ABE is formulated and addressed. Two kinds of key abuse problems are considered, i) illegal key sharing among colluding users and ii) misbehavior of the semi-trusted attribute authority including illegal key (re-)distribution. Both problems are extremely important as in an ABE-based access control system, the attribute private keys directly imply users' privileges to the protected resources. To the best knowledge of ours, such key abuse problems exist in all current ABE schemes as the attribute private keys assigned to the users are never designed to be linked to any user specific information except the commonly shared user attributes.

To be concrete, we focus on the prevention of key abuse in CP-ABE in this paper [3]. The notion of accountable CP-ABE (CP-A^{2}BE, in short) is first proposed to prevent illegal key sharing among colluding users. The accountability for user is achieved by embedding additional user specific information in the attribute private key issued to the user. To further obtain accountability for the attribute authority as well, the notion of strong CP-A^{2}BE is proposed, allowing each attribute private key to be linked to the corresponding user's secret that is unknown to the attribute authority. We show how to construct such a strong CP-A^{2}BE and prove its security based on the computational Diffie-Hellman assumption. Finally, we show how to utilize the new technique to solve some open problems existed in the previous accountable identity-based encryption schemes.

Keywords: Access control, Key abuse, Attribute-based, Ciphertext-policy, Encryption, Trace

1 Introduction

Nowadays, more and more sensitive data is shared and stored by third-party sites on the Internet. So, how to define secure and efficient access control for these data is important. Attribute-based encryption (ABE) [27] was proposed by Sahai and Waters. Recently, much attention has been attracted because ABE has been found an important application to design access control system. In ABE system, users' keys and/or ciphertexts are labeled with sets of descriptive attributes and a particular key can decrypt a particular ciphertext only if they are matched.

There are two methods for access control based on ABE: Key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). These two notions are both proposed in [21] by Goyal *et*

[3] Our technique can easily be extended to KP-ABE as well.

al. In KP-ABE, each ciphertext is labeled with sets of attributes. Each attribute private key is associated with an access structure that specifies which type of ciphertexts the key is able to decrypt. The first KP-ABE construction [21] can realize the monotonic access structures for key policies. To enable more flexible access policy, Ostrovsky *et al.* [25] presented the first KP-ABE system that supports the expression of non-monotone formulas in key policies. In a CP-ABE system, a user's key is associated with a set of attributes and an encrypted ciphertext will specify an access policy over attributes. CP-ABE is different from KP-ABE in the sense that the encryptor assigns certain access policy for the ciphertext. When a message is being encrypted, it will be associated with an access structure over a predefined set of attributes. Consequently, a user will only be able to decrypt a given ciphertext if that user's attributes pass through the corresponding access structure. Later, Bethencourt *et al.* [4] proposed the first CP-ABE construction. However, the construction [4] is only proved under the generic group model. In view of this weakness, Cheung and Newport [12] presented another construction that is proved to be secure under the standard model. The construction supports the types of access structures that are represented by AND of different attributes. Later, in [19], the authors gave another construction for more advanced access structures based on number theoretic assumption. To better protect user privacy, anonymous CP-ABE was constructed in [22] and further improved in [24]. Boneh and Waters [8] proposed a predicate encryption scheme based on the primitive called Hidden Vector Encryption. The scheme in [8] can also realize the anonymous CP-ABE by using the opposite semantics of subset predicates. Recently, Katz, Sahai, and Waters [23] proposed a novel predicate encryption scheme supporting inner product predicates and their scheme is very general and can realize both KP-ABE and hidden CP-ABE schemes. To reduce the trust of attribute authority in ABE, Chase [10] proposed a multi-authority ABE scheme, where each authority controls a subset of the attributes. If one wants to decrypt a ciphertext, he/she has to get enough attributes from every attribute authority.

Since its inaugural, ABE is envisioned as a highly promising public key primitive for realizing scalable and flexible access control systems as for the first time ABE enables public key based one-to-many encryption. It assigns differential access rights to a set of users and allow flexibility in specifying the access rights of individual users. A user has an access only if there is a match between the attributes of the ciphertext and the user's key.

However, before ABE (including both CP-ABE and KP-ABE) can be widely deployed for the purpose of providing secure access control, one important security aspect has to be properly addressed, that is, the problem of key abuse, including i) illegal key sharing among users and ii) misbehavior of attribute authority including illegal key (re-)distribution. Both problems are extremely important as in an ABE-based access control system, the attribute private keys directly imply users' privileges to the protected resources. In more details, attributes private key issued to user means the user's privileges in ABE-based access control, for example, the privilege whether the user is allowed to access to the database or not. The dishonest users may share their attribute private keys with other users without these privileges. They can just give directly away their keys, or, generate a transformed attribute private key such that nobody can tell who has distributed this. The users can even only distribute part of their privileges. If they share or give away their attribute private keys, it will make the system useless. Furthermore, we also need take the accountability of semi-trusted attribute authority into account. The attribute authority may misbehave to generate and distribute attribute private keys to users without such privileges. These problems are by far not considered in existing ABE-based access control schemes. To apply ABE for access control, we should guarantee that, 1) key issued to user cannot be shared because the key means the privilege of user; 2) attribute authority's misbehavior that distributing decryption keys or decrypting ciphertext arbitrary for users should be prevented. To the best knowledge of ours, such key abuse problems exist in all current access control schemes constructed from ABE as the attribute private keys assigned to users are never designed to be linked to any user specific information except the commonly shared user attributes. This is the reason that attribute private key can be abused by users or semi-trusted attribute authority without being detected. As a result, these protocols will be meaningless when used in access control. In this paper, we make study the problem of abuse free CP-ABE and have following contributions:

The notion of accountable CP-ABE (CP-A^{2}BE) is proposed, by inserting user specific information in the attribute private key. In CP-A^{2}BE, the accountability only for users is taken into account. To obtain further accountability for both users and attribute authority, the strong CP-A^{2}BE is proposed. To get such constructions, the technique of identity-based encryption (IBE) with wildcard [1] is utilized here.

Main Idea. The key point of the constructions is to keep the property of ABE, *i.e.*, one-to-many encryption, even if there is personal information embedded in the user's attribute private key. There are two parts in the ciphertext, 1). One part is computed through the specified attributes required in the ciphertext; 2). The other part is computed for all users with the specified attributes. The technique of IBE with wildcard is used here [1] to realize one-to-many encryption, *i.e.*, all users with the satisfied attributes could decrypt the ciphertext.

- In CP-A^{2}BE, the user's identity is embedded in the attribute private key issued to him or her from the attribute authority. The CP-A^{2}BE can be used to prevent the key sharing among users based on the following observation. In CP-A^{2}BE, the user's decryption key consists of the attribute private key and the user's identity. If the user shares its decryption key, the user's identity will be detected from the pirate device embedded with the shared decryption key, and the user will be punished.
- In strong CP-A^{2}BE, the accountability of attribute authority is achieved, apart from the accountability for users. The strong CP-A^{2}BE requires the assumption that the user should have a higher level secret than attribute private key, such as a valid public key certificate (We just consider the public key certificate for simple in this work). That is to say, the user should get a certificate for his/her public key in advance. With this public key certificate, the user could be issued attribute private key from the attribute authority, where the user's public key is embedded. Finally, define that the user's decryption key consists of the attribute private key and user's secret key corresponding to the public key certificate. If the user shares his/her decryption key, his/her secret key in the public key certificate will be leaked to others. So, this method can be utilized to prevent key sharing based on the assumption that the user would not share its secret key in the public key certificate. Moreover, the accountability of semi-trusted attribute authority can also be obtained because the user's decryption key contains some user's secret unknown to the attribute authority.

Organization. Some preliminaries are given in Section 2. We also show how to unify the definitions and security models for CP-ABE and KP-ABE. In Section 3, we propose the notion of CP-A^{2}BE and strong CP-A^{2}BE to prevent key abuse. In Section 4, the implementation and efficiency analysis are given for the strong CP-A^{2}BE. In Section 5, we show two interesting applications from the techniques used in the construction of (strong) CP-A^{2}BE. As the first application, a new accountable IBE is proposed to solve some open problems in [18]. The second application is the construction of conditional IBE. This paper ends with some concluding remarks.

2 Preliminaries

We now give a brief revision on the property of pairings and some candidates of hard problem from pairings.

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic groups of prime order p, writing the group action multiplicatively. Let g be a generator of $\mathbb{G}_1$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a map with the following properties. *Bilinearity*: $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$ for all $g_1, g_2 \in \mathbb{G}_1$, and $a, b \in_R \mathbb{Z}_p$; *Non-degeneracy*: There exist $g_1, g_2 \in \mathbb{G}_1$ such that $\hat{e}(g_1, g_2) \neq 1$, in other words, the map does not send all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$; *Computability*: There is an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}_1$. And, we also give the definitions of the following problems and assumptions based on the bilinear groups.

CDH Problem. The Computational Diffie-Hellman (CDH) problem is that, given g, g^x, g^y $\in \mathbb{G}_1$ for unknown random $x, y \in \mathbb{Z}_p^*$, to compute g^{xy}.

We say that the (t, ϵ)-CDH assumption holds in $\mathbb{G}_1$ if no t-time algorithm has the probability at least ϵ in solving the CDH problem for non-negligible ϵ.

DBDH Problem. The Decision Bilinear Diffie-Hellman (DBDH) problem is that, given g, g^x, g^y, $g^z \in \mathbb{G}_1$ for unknown random $x, y, z \in \mathbb{Z}_p^*$, $T \in \mathbb{G}_2$, to decide if $T = \hat{e}(g, g)^{xyz}$.

We say that a polynomial-time adversary $\mathcal{A}$ has advantage ϵ in solving the DBDH problem in groups $(\mathbb{G}_1, \mathbb{G}_2)$ if $\mid Pr[\mathcal{A}(g, g^x, g^y, g^z, \hat{e}(g, g)^{xyz}) = 1] - Pr[\mathcal{A}(g, g^x, g^y, g^z, \hat{e}(g, g)^r) = 1] \mid$ $\geq 2\epsilon$, where the probability is taken over the randomly chosen x, y, z, r and the random bits consumed by $\mathcal{A}$. (t, ϵ)-DBDH assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if no t-time algorithm has the probability at least ϵ in solving the DBDH problem for non-negligible ϵ.

2.1 Syntax

Next, we show the definitions for CP-ABE and KP-ABE can be unified as X-ABE, where X means CP or KP here. A binary relation R is also defined here according to concrete requirements to unify CP-ABE and KP-ABE. Denote by $R(L, W) = 1$ if L and W satisfy the relation R. Here L and R represent general access structure (key-policy) and ciphertext-policy, respectively.

Definition 1. *An X-ABE system consists of four algorithms, namely, Setup, KeyGen, Enc, and Dec, which are defined as follows:*

Setup(1^λ). The setup algorithm, on input security parameter 1^λ, outputs a master secret key sk and public key pk.

KeyGen(L, sk). The key generation algorithm, takes as input key-policy L and sk, outputs sk_L as the attribute private key for L.

Enc(M, W, pk). The encryption algorithm, on input a message M together with ciphertext-policy W, outputs $\mathcal{C}$, which is the encryption of M for W.

Dec(W, $\mathcal{C}$, sk_L). The decryption algorithm, takes as input the ciphertext $\mathcal{C}$ for ciphertext-policy W and the attribute private key sk_L. If $R(L, W) = 1$ for some relation R, it outputs M. Otherwise, it returns $\perp$.

Definitions for KP-ABE and CP-ABE can be derived from the above generalized definition.

1. *It is the definition for CP-ABE, if the key-policy L is just a set of attributes (or, attribute list), W denotes general ciphertext-policy, and $R(L, W) = 1$ (i.e., L satisfies W).*

2. *It is the definition for KP-ABE, if L is defined as an access structure (key-policy), W is a set of attributes, and $R(L, W) = 1$ (W matches L).*

As defined in [4, 12, 21], the security requirement for X-ABE is indistinguishability against chosen message attack (IND-CPA). The formal definition is given based on the following IND-CPA game involving an adversary $\mathcal{A}$.

Game IND-CPA

- *Setup. Choose a sufficiently large security parameter 1^λ, and run Setup to get a master secret key sk and public key pk. Retain sk and give pk to $\mathcal{A}$;*
- *Phase 1. $\mathcal{A}$ can perform a polynomially bounded number of queries to key generation oracle on key-policy L;*
- *Challenge. $\mathcal{A}$ outputs challenge W^* and two messages M_0, M_1 on which it wishes to be challenged. The challenger randomly chooses a bit $b \in \{0, 1\}$, computes $\mathcal{C} = Enc(M_b, W^*, pk)$ and sends $\mathcal{C}$ to $\mathcal{A}$;*
- *Phase 2. $\mathcal{A}$ continues to issue queries to the key generation oracle;*
- *Guess. Finally, $\mathcal{A}$ outputs a guess bit b'.*

$\mathcal{A}$ wins the game if $b = b'$ and L that satisfies $R(L,W^*)$=1 has not been submitted to key generation oracle. The advantage of $\mathcal{A}$ in Game IND-CPA is defined as the probability that $\mathcal{A}$ wins the game minus 1/2.

As described above, if the key-policy L is just a set of attributes (or, attribute list) and W denotes general ciphertext-policy, it is the security definition for CP-ABE. If L is defined as general access structure (key-policy) and W is defined as a set of attributes, then it is the security definition for KP-ABE.

Definition 2. *An X-ABE satisfies IND-CPA if no polynomial time adversary can break the above game.*

In this work, we will also use another weaker security model, called selective-IND-CPA in X-ABE. This model can be considered to be analogous to the selective-ID model [5] utilized in IBE protocols. In this security model, the adversary should commit to the challenge W^* before *Setup* phase.

3 Strong CP-A^{2}BE

3.1 The CP-A^{2}BE

In CP-A^{2}BE, as explained, we consider how to obtain accountability for users. The definition for CP-A^{2}BE is almost the same with CP-ABE, except here the algorithm for tracing is added.

Trace. *This algorithm is used to trace a decryption key to its original holder. It takes as input a well-formed decryption key, and outputs identity associated with this decryption key.*

Because the CP-A^{2}BE is still one kind of CP-ABE, the security requirement of IND-CPA is also necessary here. The definition of IND-CPA is almost the same with its definition in CP-ABE. To trace the identity who shares the decryption key, security requirement of accountability for users is defined here additionally. This kind of security means that if a user has decryption key for identity ID on some attributes, it cannot output new decryption key for a different identity. The accountability is defined through the following game of Key Unforgeability. This security definition is reasonable because the user has to share a decryption key with different identity from its own, to escape to be traced. In this work, we will consider a weaker security notion called selective-key unforgeability. In CP-A^{2}BE, it specifically assumes that the decryption key is well-formed because the user has to share it with others. This assumption also has been used by Goyal in [18] to reduce the trust of PKG in IBE. As mentioned in [18], like PKG in IBE, the user here could also construct a malformed decryption key which, when used in conjunction with some other decryption process, is still able to decrypt ciphertexts. So, we also need to consider the extreme case of black box which is able to decrypt the ciphertexts in practice. It is very similar to the technique of black-box traitor tracing [13]. To fix this problem and realize the identification in access control, we can transform the non-interactive to interactive, which can be easily ensure the decryption key is well-formed. So, in this paper, we only assume the decryption key is well-formed, that is, it is able to pass the key-sanity check by user. The formal definition for selective-Key Unforgeability is based on the following game involving an adversary $\mathcal{A}$.

Game selective-Key Unforgeability

- *Initial. The adversary outputs the target identity ID^* before setup;*
- *Setup. Choose a sufficiently large security parameter 1^λ, and run Setup to get a master secret key sk and public key pk. Retain sk and give pk to $\mathcal{A}$;*
- *Query. $\mathcal{A}$ can perform a polynomially bounded number of queries to key generation oracle for private key on (ID, L);*
- *Forge. Finally, $\mathcal{A}$ outputs a decryption key sk_{ID^*,L^*} for identity ID^* on attributes L^*. The challenger runs a sanity check on sk_{ID^*,L^*} to ensure that it is well-formed. It aborts if the check fails.*

$\mathcal{A}$ wins the game if ID^* has not been submitted to key generation oracle. The advantage of $\mathcal{A}$ in Game selective-Key Unforgeability is defined as the probability that $\mathcal{A}$ wins the game. The CP-A^{2}BE is accountable if there is no adversary wins the above game with non-negligible probability.

Next, we give a CP-A^{2}BE construction, which has the same access structure (ciphertext-policy) with CP-ABE scheme [12]. Details of the access structure in [12] are described below.

Assume that the total number of attributes in the system is n and the universal attributes set is $U = \{w_1, w_2, \cdots, w_n\}$. To encrypt a message, it specifies the ciphertext-policy $W = [W_1, W_2, \cdots, W_n]$. The notion of wildcard $*$ in the ciphtertext policies means the value of "don't care". For example, when $n = 4$ and let the ciphertext-policy $W = [1, 0, 1, *]$. This ciphertext policy means that the recipient who wants to decrypt must have the value 1 for W_1 and W_3, the value 0 for W_2, and the value for W_4 can be any possible value for this attribute. So, if the receiver has the attribute private keys for list $[1, 0, 1, 0]$, it can decrypt the ciphertext, because the first three values for W_1, W_2 and W_3 are equal to the corresponding values in ciphertext policy. Moreover, the fourth value 0 for W_4 satisfies the ciphertext-policy in the ciphertext because it is $*$. But if an attribute private key is associated with list $[1, 1, 1, 0]$, then, it can not decrypt because the value for the second attribute is not the same with 0 in W_2. To be more generalized, given an attribute list $L = [L_1, L_2, \cdots, L_n]$ and a ciphertext-policy $W = [W_1, W_2, \cdots, W_n]$, we say L matches W if for all $i = [1, n]$, $L_i = W_i$ or $W_i = *$. We also use the notation $R(L, W) = 1$ to indicate that L matches W. In [12], each attribute can take two values 1 and 0. In this construction, we generalize the access structures such that each attribute can take two or more values and each w_i in U can be any subset of possible values. More formally, let $S_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,n_i}\}$ be a set of possible values for w_i where n_i is the number of the possible values for w_i. Then the attribute list L for a user is $L = [L_1, L_2, \cdots, L_n]$ where $L_i \in S_i$ for $1 \leq i \leq n$, and the generalized ciphertext policy W is $W = [W_1, W_2, \cdots, W_n]$. The attribute list L satisfies the ciphertext policy W if $L_i = W_i$ or $W_i = *$ for $1 \leq i \leq n$. The construction is similar to IBE with wildcard [1] to some extend. However, the hierarchical property is not required here, which allows the construction to be more efficient and, even does not rely on random oracle model.

Let $\mathbb{G}_1, \mathbb{G}_2$ be cyclic groups of prime order p, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a pairing defined in Section 2. Assume there are n attributes in universe. That is, let the universal attributes set be $U = \{\omega_1, \omega_2, \cdots, \omega_n\}$. And, each attribute has multiple values, where S_i is the multi-value set for ω_i. The CP-A^{2}BE construction is as follows:

Setup. To generate system parameters, a trusted authority selects random generators g, g_2, g_3, u_0, u_1, ..., $u_n \in \mathbb{G}_1$, picks a random $\alpha \in \mathbb{Z}_p$, and sets $g_1 = g^\alpha$. Define a cryptographic hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*$. The system parameter is $param = (g, g_1, g_2, g_3, u_0, u_1, \cdots, u_n, H)$ and the master secret key for attribute authority is g_2^α.

KeyGen. Let $L = [L_1, L_2, \cdots, L_n]$ be the attribute list for the user with identity $ID \in \mathbb{Z}_p$. The attribute authority picks up a random $r \in \mathbb{Z}_p$ and computes the attribute private key for ID on L as (d_0, d_1), where $d_0 = g_2^\alpha (u_0^{ID} u_1^{H(L_1)} \cdots u_n^{H(L_n)} \cdot g_3)^r$, $d_1 = g^r$. The validity of (d_0, d_1) can be verified through the following equation: $\hat{e}(d_0, g) = \hat{e}(g_1, g_2)\hat{e}(u_0^{ID} u_1^{H(L_1)} \cdots u_n^{H(L_n)} \cdot g_3, d_1)$. Finally, the user retains the decryption key $sk_{ID,L} = (d_0, d_1, d_2, d_3)$ on decryption device, where $d_2 = ID$ and $d_3 = L$.

Enc. To encrypt a message $M \in \mathbb{G}_2$ under ciphertext-policy $W = [W_1, W_2, \cdots, W_n]$, pick up a random value $s \in \mathbb{Z}_p$ and set $C_0 = M\hat{e}(g_1, g_2)^s$, $C_1 = g^s$, $C_2 = (\prod_{W_i \neq *} u_i^{H(W_i)} \cdot g_3)^s$, $T_i = \{u_i^s\}_{W_i = *}$, $E = u_0^s$. The ciphertext for M with respect to W is $C = (C_0, C_1, C_2, \{T_i\}_{W_i = *}, E)$.

Dec. To decrypt the ciphertext $C = (C_0, C_1, C_2, \{T_i\}_{W_i = *}, E)$, the recipient with identity ID and attribute list L can check W to know whether $R(L, W) = 1$. If $R(L, W) = 1$, proceed as follows: Let $sk_{ID,L} = (d_0, d_1, d_2, d_3)$ be the decryption key deposited in decryption device, where $d_2 = ID$ and $d_3 = L$. The recipient first computes

$$C_2' = C_2 \prod_{W_i = *} T_i^{H(L_i)} E^{ID}$$

and decrypts the ciphertext by using $sk_{ID,L}$ as $M = C_0 \frac{\hat{e}(d_1, C_2')}{\hat{e}(d_0, C_1)}$.

Trace. Let $sk_{ID,L} = (d_0, d_1, d_2, d_3)$ be a well-formed decryption key in illegal decryption device shared by some user, where $d_3 = [L_1, L_2, \cdots, L_n]$. It means that $\hat{e}(d_0, g) = \hat{e}(g_2, g_1)$ $\hat{e}(u_0^{d_2} u_1^{H(L_1)} \cdots\ u_n^{H(L_n)}\ g_3,\ d_1)$. Then, just reveal d_2 as the identity of the dishonest user.

We have the following security result for the above construction:

Theorem 1. *The CP-A^{2}BE construction is secure in selective-IND-CPA and selective-Key Unforgeability models, under the DBDH and CDH assumptions, respectively.*

Proof. See Appendix A.

Actually, if the hash function H is viewed as random oracle in proof, the scheme can achieve the security of full Key Unforgeability. To get IND-CCA from IND-CPA encryption, one of the most efficient transformations is Fujisaki-Okamoto technique [15], which adds only a little computation on the original CP-A^{2}BE scheme. So, the resulted IND-CCA construction is very efficient.

3.2 The Strong CP-A^{2}BE

In CP-A^{2}BE, the accountability only for users is achieved. To further reduce the trust and get accountability for attribute authority, the notion of strong CP-A^{2}BE is given in this section. This also solves partly an open question given by Goyal [18] on how to design an ABE scheme with accountability for attribute authority. To construct such a strong CP-A^{2}BE scheme, the assumption that users in this system should have a higher level secret information is required. We will just focus on the example of public key certificate as a higher level secret information in this work for simple. This assumption is reasonable because, before the user is issued attribute private key, the attribute authority should know it is the right user as allege. To authenticate, the user should give proof it is the holder of a public key certificate.

The construction is based on the scheme in Section 3.1. The advantage of this method is the tracing algorithm is not required here to construct the CP-A^{2}BE. Here, however, another trace algorithm will be required to detect the misbehavior of attribute authority. We will analyze these in more detail later in this section. As explained, it assumes the user requesting attributes has public certificate. And, the user's secret key for its corresponding public key is viewed as another default attribute in the construction.

The accountability for users is described through the following game Key Unforgeability. The game of Key Unforgeability has some difference from the same game for prevention of key sharing in Section 3.1. In that game in Section 3.1, sharing can be prevented because the identity of the user will be detected if the user shares his/her decryption key. And, the user will be punished if such sharing was found. In this game, as explained, we know the users will not share the secret key in the public key certificate. Moreover, the decryption key in this definition includes the attribute private key and secret key. Based on this, in the game of Key Unforgeability, we only need to guarantee the user can not change the secret key in his/her decryption key. Because we use the public key certificate here, before issued attribute private key, the user has to prove it knows the secret key in public key certificate by using proof of knowledge technique. In Key Unforgeability game, the adversary will try to output a decryption key with some secret key without relation to its own public key. We will also define a weaker security notion called selective-Key Unforgeability as Section 3.1. That is to say, the adversary should output the key that it will use in the forged decryption key.

Game selective-Key Unforgeability

- *Initial. The adversary outputs value sk^*, corresponding to some public key pk^* that will be shared as part of the decryption key.*
- *Setup. The challenger chooses a sufficiently large security parameter 1^λ, and runs Setup to get master key sk and public key pk. Retain the secret key sk and give pk to $\mathcal{A}$.*
- *Query. $\mathcal{A}$ can perform a polynomially bounded number of queries to key generation oracle for private key on attribute list W with valid proof of knowledge to public key pk.*
- *Forge. Finally, $\mathcal{A}$ outputs a decryption key sk_{pk^*,W^*} on attributes W^* with respect to pk^*.*

$\mathcal{A}$ wins the game if sk_{pk^*,W^*} is a well-formed valid decryption key, and the public key pk^* has not been submitted to attribute private key generation oracle (There is no requirement of certificate for this public key). The advantage of $\mathcal{A}$ in Game selective-Key Unforgeability is defined as the probability that $\mathcal{A}$ wins the game. A strong CP-A^{2}BE satisfies accountability for users if there is no adversary wins the above game with non-negligible probability. One may argue in case that, after a user get an attribute private key on attributes W with respect to pk^* (Here, to query attribute private key on pk^*, pk^* should be valid public key), the user forges and shares another valid decryption key on attributes W^* with respect to pk^*. In this case, from the decryption key, we know the pk^* and, trace the identity of the public key pk^*.

From the above two games, it can achieve the security of CP-A^{2}BE. To define the strong CP-A^{2}BE, the game FindKey should be defined additionally. The accountability for attribute authority can be guaranteed based on the following game FindKey. This game is utilized to detect the misbehavior of attribute authority.

Game FindKey

- *Initial. The challenger runs Setup to get master key sk and public key pk. It gives pk, sk to the adversary $\mathcal{A}$ ($\mathcal{A}$ plays the role of semi-trused attribute authority in this game).*
- *Detect. Finally, $\mathcal{A}$ outputs a well-formed decryption key sk_{pk^*,W^*} on some attributes W^* with respect to pk^*.*

The advantage of $\mathcal{A}$ in Game Findkey is defined as the probability that $\mathcal{A}$ outputs a well-formed decryption key with respect to some user's valid public key pk^*. A strong CP-A^{2}BE satisfies accountability for attribute authority if there is no adversary wins the above game with non-negligible probability. In case that the adversary outputs some decryption key with respect to an invalid public key, we define it loses the game according to the above definition. And, from this kind of forgery, we say that the attribute authority misbehaves. In the following construction, we assume the attribute authority uses the same group as the users' public keys in public key certificate.

Setup. This algorithm is the same with Section 3.1. The system parameter is $param = (g,\, g_1,\, g_2,\, g_3,\, u_0,\, u_1,\, \cdots,\, u_n,\, H)$ and the master key is g_2^α.

KeyGen. Let $L = [L_1, L_2,\, \cdots,\, L_n]$ be the attribute list for the user with public key $u = u_0^x$. First, the user should prove it is the holder of public key u by using proof of knowledge technique. If the proof passes, the attribute authority picks up a random $r \in \mathbb{Z}_p$ and computes $d_0 = g_2^\alpha \cdot (uu_1^{H(L_1)} \cdots u_n^{H(L_n)} g_3)^r$, $d_1 = g^r$. Then, (d_0, d_1) is sent to the user as the attribute private key. Finally, the user retains the decryption key $sk_{u,L} = (d_0, d_1, d_2, d_3)$ in decryption device, where $d_2 = x$, $d_3 = [L_1, L_2,\, \cdots,\, L_n]$. The correctness of $sk_{u,L}$ can be verified if $\hat{e}(d_0, g) = \hat{e}(g_1, g_2)\hat{e}(u_0^{d_2} u_1^{H(L_1)} \cdots u_n^{H(L_n)} \cdot g_3, d_1)$.

Enc. To encrypt a message $M \in G_2$ under a ciphertext-policy $W = [W_1, W_2, \cdots, W_n]$, it proceeds as the algorithm *Enc* in Section 3.1. Finally, it outputs the ciphertext for M with respect to W as $C = (C_0, C_1, C_2, \{T_i\}_{W_i=*}, E)$.

Dec. To decrypt the ciphertext $C = (C_0, C_1, C_2, \{T_i\}_{W_i=*}, E)$, the recipient with public key u and attribute list L can check W to know whether $L \models W$. If $L \models W$, it can proceed as follows: Let $sk_{u,L} = (d_0, d_1, d_2, d_3)$ be the decryption key. The recipient computes

$$C_2' = C_2 \prod_{W_i=*} T_i^{H(L_i)} E^{d_2}$$

and decrypts the ciphertext by using $sk_{u,L}$ as $M = C_0 \frac{\hat{e}(d_1, C_2')}{\hat{e}(d_0, C_1)}$.

The above construction can be proved to be a secure CP-A^{2}BE scheme, without the requirement of *Trace* algorithm. However, to achieve strong CP-A^{2}BE, it is also necessary to detect the misbehavior of the attribute authority. So, the algorithm *Trace* will be also given here for accountability of attribute authority.

Trace. Let $sk_{u,L} = (d_0, d_1, d_2, d_3)$ be a well-formed decryption key in illegal decryption device, where $d_3 =[L_1, L_2,\, \cdots,\, L_n]$. It means that $\hat{e}(d_0, g) = \hat{e}(g_2, g_1)\ \hat{e}(u_0^{d_2} u_1^{H(L_1)} \cdots\ u_n^{H(L_n)}\ g_3,\, d_1)$. Then, compute $u_0^{d_2}$ and check if $u_0^{d_2}$ is a valid public key. If not, then, it is the attribute authority who generates and distributes this decryption key.

Note that in *KeyGen* algorithm, the user with public key u should prove he is the holder of this public key by proving the knowledge of x. The secret key x can be viewed as another attribute issued in the attribute private key though the attribute authority does not know the attribute. So, the proof can be easily get from the proof in Section 3.1.

Theorem 2. *The strong CP-A^{2}BE construction is secure in* selective-IND-CPA *model, and has accountability for users and attribute authority, under the* DBDH *assumption and* CDH *assumption, respectively.*

Proof. We only prove that it achieves accountability for users under selective-Key Unforgeability model. First, the adversary outputs some value sk^* in private key it wants to share. Then, the simulator sets the public parameters and simulate the private key generation oracle. It is the same as in proof of Theorem 1. Notice here sk^* is viewed as ID^*. The simulator only needs to simulate the key generation oracle. The challenge ciphertext oracle is not required here because the goal of adversary is to output private key for any attributes that is not for its own public key pk. The adversary could ask private key for any W with respect to valid public key pk_u. During key generation queries, the user will not give simulator the secret key sk_u. However, as mentioned in *KeyGen*, the user has to use some proof of knowledge to show he/she is the holder of public key pk_u. By using the knowledge extractor, the simulator can extract sk_u and simulate key generation oracle the same as in Theorem 1, where ID is viewed as sk_u here. The accountability of attribute authority can be easily proved from the definition of game FindKey. In the decryption key, it includes the secret key sk_u with respect to pk_u. To avoid detecting, the attribute authority has to output a valid decryption key on user's public key. This implies the attribute authority has to compute the user's secret key corresponding to the public key in his/her public key certificate. Of course, based on the security of public key system, we can get that the above CP-A^{2}BE has accountability for attribute authority.

4 Implementation for Access Control

Assume there is some database for the storage of information. User could be allowed to access the database according to its privilege. In the access control system based on ABE, the privilege is categorized through the users' attributes.

First, each user in this system should have a public key certificate, which function as a higher level private information compared to the attribute-based system. The user chooses some secret key and computes its corresponding public key. Then, the user proves to the certificate authority that it knows the corresponding secret key in the public key by using proof of knowledge method. After the registration of public key, the user could be issued private keys for its attributes. The attribute authority is in charge of the attributes issue, including the check for privilege authentication and public key authentication. In the above construction, the group used by the attribute authority should be the same with the one used in the public key certificate. After the check passes, the user can get private key for some attributes for the corresponding privileges from the attribute authority.

To encrypt the message such that only users with special attributes can decrypt, just compute the ciphertext according to the required policy, for example, using the given ciphertext-policy above. This only allows the users, with attributes that matches the ciphertext, can decrypt the ciphertext. When the above CP-ABE is used for access control, the access policy is defined as AND, that is, the ciphertext-policy is AND of attributes. The user with some attributes would not share its key with other users because there is user's secret key, which is corresponding to public key certificate, in the attribute private key. And, the user cannot compute a new private key with a different invalid public key (This property can be achieved based on the security game of FindKey). So, if a valid attribute private key with respect to some invalid public key was found, then, we can tell it is the misbehavior of attribute authority.

We achieved the same access structure as in [12]. In our scheme, the private key size for user consists of only two group elements, which is constant with the number of user's attribute. However, in [12], there are $2n$ group elements in the private key for user. In the key generation algorithm, it performs two exponentiations in group $\mathbb{G}_1$ for the attribute authority to generate private key for any user. However, $2n$ exponentiations in $\mathbb{G}_1$ are required in key generation algorithms in [12], where n is the number of attributes in universe. The computational cost [12] is linear with the number of universal attributes. In our construction, ciphertext consists of $3+k$ group elements and the encryption algorithm needs $3+k$ exponentiations in $\mathbb{G}_1$, where k denotes the number of wildcards in ciphertext. Decryption requires two pairing computation. However, in [12], there are $3n$ group elements in ciphertext and encryption algorithm performs $n+1$ exponentiations in group $\mathbb{G}_1$. For decryption, it performs $n+1$ pairings. So, overall, the above construction is more efficient than [12].

5 Other Applications of The New Techniques

In this section, we show how to use the above technique to solve some open problems existed in accountable IBE. Then, we propose another notion called conditional IBE.

5.1 Accountable IBE

Identity-based cryptosystem [29] is a public key cryptosystem where the public key can be an arbitrary string such as an email address. It was proposed to simplify key management procedures of certificate-based public key infrastructures. In identity-based cryptosystem, a private key generator (PKG) uses a master secret key to issue private keys to identities that request them. Until 2001, Boneh and Franklin [7] proposed the first practical IBE scheme based on pairing, which is provably secure in the random oracle model. The first IBE without random oracles was proposed by Waters [30]. Later, this construction was further generalized and analyzed in [11]. To reduce the trust of PKG in IBE, Goyal [18] proposed another notion called accountable IBE and strengthened by [2, 20]. These IBE can only be used to encrypt message to a single user. To encrypt a message for a group, the notion of IBE with wildcard [1] was proposed. In this kind of IBE, one can encrypt a message for a group of users with some common properties, in which the different parts are viewed as "don't care" components. Recently, Goyal [18] proposed the first method on how to reduce the trust of PKG in IBE. Though later construction with black-box accountability based on DBDH assumption was proposed [20], it is very inefficient.

An open question was left in [18]: How to construct a more efficient IBE with minimum trust to PKG based on standard assumption such as DBDH assumption? In this section, we solve this open question by using the technique used in above sections. Our construction is more efficient than [18] and, based on DBDH assumption. We combine user's public key certificate with IBE system, while keeping all the properties of IBE. Actually, to construct some identity-based cryptosystem with special properties, [16] has proposed to use the public key certificate.

Another open problem can also be solved by using our method: In case of some user loses private key for its identity ID, then, how to achieve accountability? The papers [18, 20] cannot solve this problem because they require the user's decryption key to detect the misbehavior of PKG. In our method, even with only one private key for ID in pirate device, this kind of key abuse can be detected because the value in the forged decryption key is different from the public key in the public key certificate for ID.

Setup. This algorithm is almost the same with Section 3.1, except only part of parameters from it is required here. The system parameter is $param = (g, g_1, g_2, g_3, u_0, u_1, H)$ and the master key is g_2^{α}.

KeyGen. Let identity be ID for the user with public key $u = u_0^x$. First, the user should prove he is the holder of public key u by using the technique proof of knowledge. If the proof passes, the attribute authority picks up a random $r \in \mathbb{Z}_p$ and computes (d_0, d_1), where $d_0 = g_2^\alpha \cdot (uu_1^{ID}g_3)^r$, $d_1 = g^r$. Finally, the user retains the decryption key $sk_{ID} = (d_0, d_1, d_2)$ for decryption, where $d_2 = x$. The correctness of (d_0, d_1, d_2) can be verified by checking if the following equation holds: $\hat{e}(d_0, g) = \hat{e}(g_1, g_2)\hat{e}(u_0^{d_2}u_1^{ID}g_3, d_1)$.

Enc. To encrypt a message $M \in \mathbb{G}_2$ for user ID, pick up a random value $s \in \mathbb{Z}_p$ and set $C_0 = M\hat{e}(g_1, g_2)^s$, $C_1 = g^s$, $C_2 = (u_1^{ID}g_3)^s$, $E = u_0^s$. Then, the cipertext for M with respect to ID is $C = (C_0, C_1, C_2, E)$.

Dec. To decrypt the ciphertext $C = (C_0, C_1, C_2, E)$, the user ID with public key u proceeds as follows: Let $sk_{ID} = (d_0, d_1, d_2)$ be the decryption key for ID. The user computes $C_2' = C_2E^x$ and decrypts the ciphertext as $M = C_0\frac{\hat{e}(d_1, C_2')}{\hat{e}(d_0, C_1)}$.

Trace. Take as input a well-formed decryption key sk_{ID} =(d_0, d_1, d_2). It means that $\hat{e}(d_0, g) = \hat{e}(g_1, g_2)\hat{e}(u_0^{d_2}u_1^{ID}g_3, d_1)$. Then, output $u_0^{d_2}$ and check if $u_0^{d_2}$ is equal to u in public key certificate for ID. It means the PKG forges and distributes the decryption key for user ID if $u_0^{d_2} \neq u$.

We describe briefly why this technique can prevent PKG from generating private key and decrypting ciphertext on behalf of user. If PKG outputs a forged valid decryption key for ID, then, in the well-formed decryption key, another different value $u_0^{x'}$ will be inserted. From the public certificate, we know the public key u_0^x and identity ID is connected to the same user. So, from these two valid decryption keys, it can tell that PKG forges the decryption key for user with identity ID. In our system, even the user with identity ID has not requested private key, from the forged decryption key for ID, we can tell PKG behaves illegally because there is no certificate for public key u' and ID. But in [18], it can not prevent such forgery because we can only tell that PKG forges the decryption key if we have two different decryption keys for the same identity ID. And, fully-secure accountable IBE can be constructed, based on the Waters IBE without random oracles [30]. The technique is similar to the above construction.

5.2 Conditional IBE

Consider the following scenario: If one wants to encrypt a message to some identity ID. Moreover, apart from identity, the encryptor wants to ensure that not only the identity of the user is ID, but also the user satisfies some additional conditions. For example, the user with identity ID can decrypt the ciphertext for him if he also has attributes “Ph.D degree” and “Staff in University A”. However, in traditional IBE, the private key is only related to identity information. From the above CP-A^{2}BE scheme, such kind of IBE with additional conditions could be derived.

The *Setup* and *KeyGen* algorithms, including the definition for ciphertext-policy, are the same as the scheme in Section 3.1.

Enc. To encrypt a message $M \in \mathbb{G}_2$ for identity ID with ciphertext policy $W = [W_1, W_2, \cdots, W_n]$, pick up a random value $s \in \mathbb{Z}_p$ and set $C_0 = M\hat{e}(g_1, g_2)^s$, $C_1 = g^s$, $C_2 = (\prod_{W_i \neq *} u_0^{H(ID)}u_i^{H(W_i)}g_3)^s$, $T_i = \{u_i^s\}_{W_i=*}$. Then, the cipertext to identity ID for M with respect to W is $C = (C_0, C_1, C_2, \{T_i\}_{W_i=*})$.

Dec. To decrypt the ciphertext $C = (C_0, C_1, C_2, \{T_i\}_{W_i=*})$ for ID on W, the recipient with identity ID and attribute list $L = [L_1, L_2, \cdots, L_n]$ can check W to know whether $R(L, W) = 1$. If $R(L, W) = 1$, proceed as follows: Let $sk_{ID,L} = (d_0, d_1)$ be the secret key for L. The recipient computes

$$C_2' = C_2 \prod_{W_i=*} T_i^{H(L_i)}$$

and decrypts the ciphertext by using $sk_{ID,L}$ as $M = C_0 \frac{\hat{e}(d_1, C_2')}{\hat{e}(d_0, C_1)}$.

6 Conclusion

In this paper, we discussed the problem of key abuse existed in access control that is based on CP-ABE. Two kinds of accountability are considered in this work: The accountability for users and accountability for the semi-trusted attribute authority. First, we showed how to construct CP-A^{2}BE to achieve accountability for users, by inserting user's specific information, such as the user's identity. To obtain accountability for both users and the semi-trusted attribute authority, we proposed and formulated the notion of strong CP-A^{2}BE by letting the decryption key contain user's secret unknown to the attribute authority. A strong CP-A^{2}BE scheme was also constructed based on the assumption that each user has a public key certificate. The key point to these constructions is that the user's personal information or secret could be viewed as another default attribute.

Finally, we solved some open problems in accountable IBE by utilizing the new technique. Another application is to design the conditional IBE scheme.

References

1. Michel Abdalla, Dario Catalano, Alexander W. Dent, John Malone-Lee, Gregory Neven, and Nigel P. Smart. *Identity-Based Encryption Gone Wild.* ICALP'06, LNCS 4052, pp. 300-311, Springer, 2006.
2. Man Ho Au, Qiong Huang, Joseph K. Liu, Willy Susilo, Duncan S. Wong, and Guomin Yang. *Traceable and Retrievable Identity-Based Encryption.* ACNS'08, LNCS 5037, pp. 94-110, Springer, 2008
3. Joonsang Baek, Willy Susilo, and Jianying Zhou. *New Constructions of Fuzzy Identity-Based Encryption.* ASIACCS'07, pp. 368-370, ACM, 2007.
4. John Bethencourt, Amit Sahai, and Brent Waters. *Ciphertext-Policy Attribute-Based Encryption.* IEEE Symposium on Security and Privacy'07, pp. 321-334, IEEE, 2007.
5. Dan Boneh and Xavier Boyen. *Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles.* EUROCRYPT'04, LNCS 3027, pp. 223-238, Springer, 2004.
6. Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. *Chosen-Ciphertext Security from Identity-Based Encryption.* SIAM J. Comput. 36(5): 1301-1328, 2007.
7. Dan Boneh and M. Franklin. *Identity-Based Encryption from The Weil Pairing*, Crypto'01, LNCS 2139, pp.213-229, Springer-Verlag, 2001.
8. Dan Boneh and Brent Waters. *Conjunctive, Subset, and Range Queries on Encrypted Data.* TCC'07. LNCS 4392, pp. 535-554. Springer, 2007.
9. Xavier Boyen, Qixiang Mei, and Brent Waters. *Direct Chosen Ciphertext Security from Identity-Based Techniques.* CCS'05, pp. 320-329, ACM press, 2005. Full version at http://eprint.iacr.org/2005/288.
10. Melissa Chase. *Multi-Authority Attribute Based Encryption.* TCC'07, LNCS 4392, pp. 515-534, Springer, 2007.
11. Sanjit Chatterjee and Palash Sarkar. *Trading Time for Space: Towards an Efficient IBE Scheme with Short(er) Public Parameters in the Standard Model*, ICISC'05, LNCS 3935, pp. 424-440, 2005.
12. Ling Cheung and Calvin Newport. *Provably Secure Ciphertext Policy ABE.* In CCS'07, Proceedings of the 14th ACM conference on Computer and communications security, pages 456-465, ACM, 2007.

13. Benny Chor, Amos Fiat, and Moni Naor. *Tracing Traitor.* CRYPTO'94, LNCS 839, pp. 257-270, Springer, 1994.
14. Paul Feldman. *A Practical Scheme for Non-Interactive Verifiable Secret Sharing.* In Proc. 28th FOCS, pp. 427-437, 1987.
15. Eiichiro Fujisaki and Tatsuaki Okamoto. *Secure Integration of Asymmetric and Symmetric Encryption Schemes.* CRYPTO'99, LNCS 1666, pp. 537-554, Springer, 1999.
16. David Galindo, Javier Herranz, and Eike Kiltz. *On the Generic Construction of Identity-Based Signatures with Additional Properties.* ASIACRYPT'06, LNCS 4284, pp. 178-193, Springer, 2006.
17. Craig Gentry. *Practical Identity-based Encryption without Random Oracles.* EUROCRYPT'06. LNCS, vol. 4004, pp. 445-464. Springer, 2006.
18. Vipul Goyal. *Reducing Trust in the PKG in Identity Based Cryptosystems.* CRYPTO'07, LNCS 4622, pp. 430-447, 2007. Extension is available at http://eprint.iacr.org/2007/368.
19. Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. *Bounded Ciphertext Policy Attribute Based Encryption.* ICALP'08. LNCS 5126, pp. 579-591, 2008.
20. Vipul Goyal, Steve Lu, Amit Sahai, and Brent Waters. *Black-Box Accountable Authority Identity-Based Encryption.* CCS'08, USA.
21. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data.* CCS'06, pp. 89-98, ACM, 2006.
22. Apu Kapadia, Patrick P. Tsang, and Sean W. Smith. *Attribute-based Publishing with Hidden Credentials and Hidden Policies.* In Proc. of Network and Distributed System Security Symposium (NDSS), pp. 179-192, 2007.
23. Jonathan Katz, Amit Sahai, and Brent Waters. *Predicate Encryption Supporting Disjunctions, Polynomial Equations, and Inner Products.* EUROCRYPT'08, LNCS 4965, pp. 146-162, Springer, 2008.
24. Takashi Nishide, Kazuki Yoneyama, and Kazuo Ohta. *ABE with Partially Hidden Encryptor-Specified Access Structure.* ACNS'08, LNCS 5037, pp. 111-129, Springer, 2008.
25. Rafail Ostrovsky, Amit Sahai, and Brent Waters. *Attribute-based Encryption with Non-Monotonic Access Structures.* CCS'07, pp. 195-203, ACM, 2007.
26. Amit Sahai. *Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen Ciphertext Security.* IEEE Symp. on Foundations of Computer Science, 1999.
27. Amit Sahai and Brent Waters. *Fuzzy Identity-Based Encryption.* EUROCRYPT'05, LNCS 3494, pp. 457-473, Springer, 2005.
28. Adi Shamir. *How to Share a Secret.* vol. 22, pp. 612-613, ACM, 1979.
29. Adi Shamir. *Identity-Based Cryptosystems and Signature Schemes.* CRYPTO'84, LNCS 196, pp. 47-53, Springer, 1984.
30. Brent Waters. *Efficient Identity-Based Encryption Without Random Oracles.* EUROCRYPT'05, LNCS 3494, pp. 114-127, Springer, 2005.

Appendix A: Proof of Theorem 1

The proof of Theorem 1 can be derived from the following two Lemmas.

Lemma 1. *The CP-A^2BE is* selective-IND-CPA *secure under the DBDH assumption.*

Proof. Assume that an attacker $\mathcal{A}$ breaks selective-IND-CPA with probability greater than ϵ within time t making q_d private key extraction queries. We show that using $\mathcal{A}$, one can construct a DBDH attacker $\mathcal{A}'$ with almost the same probability with ϵ.

Initial stage. First, $\mathcal{A}$ outputs the target ciphertext-policy $W^*{=}(W_1^*, \cdots, W_n^*)$.

Setup. Suppose that $\mathcal{A}'$ is given $(g, \hat{e}, \mathbb{G}_1, \mathbb{G}_2,\ A = g^x,\ B = g^y,\ C = g^z,\ T)$, where T is either $\hat{e}(g, g)^{xyz}$ or $\hat{e}(g, g)^{\gamma}$ for random $\gamma \in \mathbb{Z}_p$, as an instance of the DBDH problem. By ϵ' and t', we denote the winning probability and running time of $\mathcal{A}'$, respectively. $\mathcal{A}'$ can simulate the challenger's execution of each phase of selective-IND-CPA game for $\mathcal{A}$ as follows: $\mathcal{A}'$ sets $g_1 = g^x$ and $g_2 = g^y$. For $1 \le i \le n$ and $W_i^* \neq *$, let $u_i = g_1^{a_i} g^{b_i}$ by choosing

$a_i, b_i \in Z_p^*$. For $1 \leq i \leq n$ and $W_i^* = *$, $u_i = g^{b_i}$ by choosing $b_i \in Z_p^*$. Then, choose a random $b_0 \in Z_p^*$ and let $u_0 = g^{b_0}$. Assign $g_3 = g_1^{-\sum_{1\leq i\leq n \wedge W_i^* \neq *} a_i H(W_i^*)} g^{b'}$. The system parameters $para$= $(g, g_1, g_2, g_3, (u_i)_{0\leq i\leq n})$ are sent to $\mathcal{A}$.

Phase 1. $\mathcal{A}'$ answers $\mathcal{A}$'s key generation queries as follows. Upon receiving a key generation query for L=$(L_1, \cdots, L_n)$ with respect to ID, $\mathcal{A}'$ checks if $L \models W^*$. If $L \models W^*$, $\mathcal{A}'$ aborts.

Otherwise, $\mathcal{A}'$ chooses $r = \frac{-y}{\sum_{1\leq i\leq n} a_i H(L_i) - \sum_{W_i^* \neq *} a_i H(W_i^*)} + r'$.

Let $R = \sum_{1\leq i\leq n} a_i H(L_i) - \sum_{W_i^* \neq *} a_i H(W_i^*)$ and $R' = \frac{-b_0 ID - \sum_{1\leq i\leq n} b_i H(L_i) + b'}{R}$.

It outputs the simulated private key as
$sk_{ID,L}$=(a_0, a_1)=$((u_0^{ID} u_1^{H(L_1)} \cdots u_n^{H(L_n)} g_3)^{r'} g_2^{R'}, g_2^{\frac{-1}{R}} g^{r'})$.

First, we need to check if $a_0 = g_2^x (u_0^{ID} u_1^{H(L_1)} \cdots u_n^{H(L_n)} \cdot g_3)^r$ and $a_1 = g^r$.

Because $R' = \frac{-b_0 ID - \sum_{1\leq i\leq n} b_i H(L_i) + b'}{R}$ and $r = \frac{-y}{R} + r'$, we have,

$$\begin{aligned} g_2^x (u_0^{ID} u_1^{H(L_1)} \cdots u_n^{H(L_n)} \cdot g_3)^r &= g_2^x (g_1^R g^{b_0 ID + \sum_{1\leq i\leq n} b_i H(L_i) + b'})^r \\ &= g_2^x (g_1^R g^{b_0 ID + \sum_{1\leq i\leq n} b_i H(L_i) + b'})^{\frac{-y}{R} + r'} \\ &= g_2^x (g_1^R g^{b_0 ID + \sum_{1\leq i\leq n} b_i H(L_i) + b'})^{r'} g^{-xy} g_2^{R'} \\ &= (g_1^R g^{b_0 ID + \sum_{1\leq i\leq n} b_i H(L_i) + b'})^{r'} g_2^{R'} \\ &= (u_0^{ID} u_1^{H(L_1)} \cdots u_n^{H(L_n)} g_3)^{r'} g_2^{R'} \end{aligned}$$

And, g^r=$g^{\frac{-y}{R} + r'}$ =$g_2^{\frac{-1}{R}} g^{r'}$.

Challenge. $\mathcal{A}$ outputs two equal length messages M_0, M_1, identity ID^*, and the challenge ciphertext-policy W^*=$(W_1^*, \cdots, W_n^*)$. It chooses randomly $b \in \{0,1\}$, and outputs the ciphertext as $(TM_b$, C, $C^{\sum_{W_i^* \neq *} b_i H(W_i^*) + b'}$, $\{C^{b_i}\}_{W_i^* *}$, $C^{b_0})$. It could be verified the ciphertext is correct if $T = \hat{e}(g,g)^{xyz}$, because $(TM_b$, C, $C^{\sum_{1\leq i\leq n, W_i^* \neq *} b_i H(W_i^*) + b'}$, $\{C^{b_i}\}_{W_i^* *}$, $C^{b_0})$=$(e(g_1, g_2)^s M_b$, g^s, C_2=$(\prod_{i=1, W_i^* \neq *}^n u_i^{H(W_i^*)} \cdot g_3)^s$, $T_i = \{u_i^s\}_{W_i = *}$, $u_0^s)$ by just letting $s = z$.

Phase 2.$\mathcal{A}$ can still query key generation. $\mathcal{A}'$ answers key generation queries as above.

Guess Finally, $\mathcal{A}$ outputs a bit b'. Then, $\mathcal{A}'$ also outputs b' as the answer to the DBDH problem. For the simulation to complete without aborting. It is easy to verify that we can get the probability of breaking the DBDH problem as $\epsilon' \approx \epsilon$ if the adversary successes with probability ϵ.

Lemma 2. *The CP-A^{2}BE is selective-Key Unforgeability under the CDH assumption.*

Proof. Assume that an attacker $\mathcal{A}$ breaks selective-Key Unforgeability with probability greater than ϵ within time t making q_d private key generation queries. We show that using $\mathcal{A}$, one can break the CDH problem by constructing another attacker $\mathcal{A}'$ with approximately the same success probability.

Initial. First, $\mathcal{A}$ outputs the target identity ID^*.

Setup. Suppose that $\mathcal{A}'$ is given $g, \hat{e}, \mathbb{G}_1, \mathbb{G}_2$, $A = g^x$, $B = g^y$ and asked to compute g^{xy}.

$\mathcal{A}'$ can simulate the challenger's execution of each phase for $\mathcal{A}$ as follows: $\mathcal{A}'$ sets $g_1 = A$ and $g_2 = B$. It chooses $r_0, r_0', r_1, r_2, \cdots, r_n \in \mathbb{Z}_p^*$. Let $u_0 = A^{r_0}$ and $g_3 = A^{-ID^* r_0} g^{r_0'}$. For $1 \leq i \leq n$, let $u_i = g^{r_i}$. The system parameters $para$= $(g, g_1, g_2, g_3, u_0, (u_i)_{1 \leq i \leq n})$ are sent to $\mathcal{A}$.

Query. $\mathcal{A}'$ answers $\mathcal{A}$'s key generation queries as Lemma 1. Upon receiving a key generation query for ID with attributes $L = [L_1, L_2, \cdots, L_n]$, $\mathcal{A}'$ chooses $r' \in \mathbb{Z}_p^*$ and lets $r = \frac{y}{(ID^* - ID) r_0} + r'$. The private key can be simulated private key as

$(g_1^{r_0(ID - ID^*)} g^{\sum_{i=1}^n r_i H(L_i)})^{r'} g_2^{r_0' + \sum_{i=1}^n r_i H(L_i)/(ID - ID^*) r_0}$, $g_2^{\frac{1}{(ID - ID^*) r_0}} g^{r'})$.

The correctness can be verified as the same way in Lemma 1.

Forgery. Finally, $\mathcal{A}$ outputs a forged decryption key $sk_{ID^*, L^*} = (d_0, d_1, d_2, d_3)$ that $\mathcal{A}$ will share for attribute list $d_3 = L^* = [L_1^*, L_2^*, \cdots, L_n^*]$ on identity $d_2 = ID^*$. Because the decryption is valid and well-formed, then, we have $d_0 = g_2^x (u_0^{H(d_2^*)} u_1^{H(L_1^*)} \cdots u_n^{H(L_n^*)} \cdot g_3)^r$ and $d_1 = g^r$ for some r.

Because the select of public parameter in setup, we could have that $d_0 = g_2^x (g^{r_0'} g^{\sum_{i=1}^n r_i H(L_i^*)})^r$. Then, $\mathcal{A}'$ can compute $g_2^x = d_0 / (d_1)^{r_0' + \sum_{i=1}^n r_i H(L_i^*)}$ and output it as the solution to the CDH problem.