Enhancing Attribute-Based Encryption with Attribute Hierarchy

Jin Li, Qian Wang, Cong Wang, and Kui Ren
Department of ECE, Illinois Institute of Technology
Email: {jin.li, qwang, cwang, kren}@ece.iit.edu

***Abstract*—Attribute-based encryption (ABE) has been envisioned as a promising cryptographic primitive for realizing secure and flexible access control. However, ABE is being criticized for its high scheme overhead as extensive pairing operations are usually required. In this paper, we focus on improving the efficiency of ABE by leveraging a previously overlooked fact, i.e., the often-found hierarchical relationships among the attributes that are inherent to many access control scenarios. As the first research effort along this direction, we coin the notion of hierarchical ABE (HABE), which can be viewed as the generalization of traditional ABE in the sense that both definitions are equal when all attributes are independent. We further give a concrete HABE construction considering a tree hierarchy among the attributes, which is provably secure. More importantly, our construction exhibits significant improvements over the traditional ABE when attribute hierarchies exist.**

Keywords: Attribute-Based, Hierarchy, Encryption, Access Control, Pairing

I. Introduction

Recently, much attention has been attracted by a new public-key primitive called Attribute-based encryption (ABE). ABE has significant advantage over the traditional PKC primitives as it achieves flexible one-to-many encryption instead of one-to-one. ABE thus is envisioned as an important tool for addressing the problem of secure and fine-grained data sharing and access control. In ABE, the encryption keys and/or ciphertexts are labeled with sets of descriptive attributes defined for the system users. And a particular user private key can decrypt a particular ciphertext only if the two match. A party could encrypt a document to all users who have a certain set of attributes drawn from a pre-defined attribute universe. For example, one can encrypt a tenure-evaluation related document to all of tenured faculty in computer science department. In this case the document would be encrypted to the attribute subset {"Faculty", "CS Dept.", "Tenured"}, and only users with all of these three attributes in the university can hold the corresponding private keys and thus decrypt the document, while others cannot.

ABE, on the other hand, is often being criticized for its high scheme overhead as extensive pairing operations are usually required. In this paper, we focus on improving the efficiency of ABE by leveraging a previously overlooked fact, i.e., the often-found hierarchical relationships among the attributes that are inherent to many access control scenarios. The notion of HABE is proposed in this paper to address the tree hierarchy structure, which can be viewed as the generalization of traditional ABE in the sense that both definitions are equal when all attributes are independent. In HABE, the universal attributes are classified into trees according to their relationship defined in the access control system. Every node in this tree is associated with an attribute, and an ancestral node can derive its descendant's key, but the reverse is not allowed. Assume the attributes form n trees. For attribute ω, we assume its depth is k in the i-th tree, and its path from root ω_{i0} in the i-th tree is defined as $(\omega_{i0}, \omega_{i1}, \cdots, \omega_{i,k-1}, \omega_{ik})$, where $\omega_{ik} = \omega$[1]. We say that ω covers ω' with path $(\omega'_{j0}, \omega'_{j1}, \cdots, \omega'_{jk'})$ if $\omega_{i\delta} = \omega'_{j\delta}$ for $0 \leq \delta \leq k$. It means that ω has higher level priority than ω' in the access control system if ω covers ω'. For convenience, we first define some notations. Recall that we wish to create an HABE scheme in which a ciphertext can be decrypted only by users with the following property: the number of users' attributes that cover the attributes included in ciphertext is no less than a pre-defined number d. Before decryption, the user can get an attribute set U from the attribute center. Assume the ciphertext is created with respect to an attribute set U'. The user with U is able to decrypt the ciphertext for U' if and only if the number of attributes in U that cover U' is no less than d. This kind of ABE could be used in distributed systems so that a user is able to access data only if he or she possesses a certain set of credentials or attributes. To construct such ABE directly without taking advantage of the hierarchical structure, the size of private key or the ciphertext will grow linearly with the number of decedents or depth of the attribute. In our HABE, part of attributes are allowed to have hierarchical tree relationship and the remaining attributes are independent. Therefore, our construction can achieve both flexibility and practicality.

A. Related Work

ABE is one of the important applications of fuzzy identity-based encryption (fuzzy IBE, in short) [23] proposed by Sahai and Waters. In fuzzy IBE, the identity is viewed as a set of descriptive attributes. A user with secret key for ω is able to decrypt a ciphertext encrypted with ω' if and only if ω and ω' are within a certain distance of each other as judged by some metric. As for [23], this distance is measured by set-overlap between identities. Due to the error-tolerance property, fuzzy IBE can be applied to enable encryption by using

[1] In this paper, assume ω is in depth k_i of the i-th tree and let its path be $(\omega_{i0}, \omega_{i1}, \cdots, \omega_{i,k-1}, \omega)$. For convenience, we will use the notation ω_{ik} instead of ω to denote its position in its path, without especial explanation in the following sections.

biometric inputs as identities. To reduce the trust of attribute authority, Chase [9] proposed a multi-authority attribute-based encryption scheme. In this protocol, each authority controls some of the attributes, and this scheme can also be extended to support tree-structure [16]. Recently, there are several attempts to construct attribute-based signature in both [19], [20]. They presented attribute (ring) signature to achieve signer privacy. These constructions are not trivial by using technique in [12] since the anonymity for user is required.

There are two methods for access control based on ABE: Key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). Both notions are proposed in [16] by Goyal *et al.* In KP-ABE, each ciphertext is labeled with sets of attributes. Each attribute private key is associated with an access structure such that it can only decrypt a specific type of ciphertext. The first KP-ABE construction [16] can realize the monotonic access structures for key policies. To enable more flexible access policy, Ostrovsky *et al.* [22] 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. Bethencourt *et al.* [3] proposed the first CP-ABE construction. However, the construction in [3] is only proved under the generic group model. In view of this weakness, Cheung and Newport [10] presented another construction that is proved to be secure under the standard model. Later, in [14], Goyal et al. gave another construction for more advanced access structures based on number theoretic assumption. To better protect user privacy, anonymous CP-ABE was constructed in [17] and further improved in [21]. Boneh and Waters [7] proposed a predicate encryption scheme based on the primitive called Hidden Vector Encryption. Their scheme can also realize the anonymous CP-ABE by using the opposite semantics of subset predicates. Recently, Katz, Sahai, and Waters [18] 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.

B. Contribution

In this paper, we make the following contributions: (i) The model of HABE is formalized; (ii) To obtain a provably secure HABE under tree hierarchy, the technique of hierarchical identity-based encryption is utilized in combination with the secret sharing techniques in ABE; (iii) We show through detailed analysis that our construction is very efficient: the computation cost in generation of ciphertext is low and the length of the ciphertext is short.

ORGANIZATION. In Section II, the model for HABE is formalized, as well as the construction. Its security analysis under the established model is also presented. In Section III, we show how to implement such HABE and give its efficiency analysis. Section IV is the concluding remarks.

II. Building Blocks: The HABE Schemes

A. Syntax

In this section, we first give the definition and security model of HABE. Then, a provably secure construction of HABE is presented. When one encrypts a message m for a set of target attributes (without loss of generality, let $\mathsf{U} = \{\omega_1, \cdots, \omega_k\}$), anyone can decrypt the ciphertext if he has at least d attributes that cover the attributes in U. The distance d should be pre-determined in setup algorithm, which will be used in the encryption and decryption algorithms. However, in some applications, the size of d is not fixed. To solve this problem, we will explain later how to make d flexible for the distance under different scenarios. The definition of HABE is similar to ordinary ABE through the definition of overlap between attributes sets, except that in HABE the attributes have hierarchical structure. It is assumed that the universal attributes form hierarchical structure according to the definition of access control system. Note that we call an attribute ω covers ω' if $\omega = \omega'$ or ω belongs to a higher level than ω'.

Definition 1: The HABE scheme consists of four algorithms (Setup, KeyGen, Enc, Dec), which are defined as follows:

- Setup: The setup algorithm takes as input security parameter 1^λ, and generates public parameters para and sk. It retains sk as the secret key for attribute center and outputs para.
- KeyGen(U, para, sk): The private key generation algorithm takes as input attribute set U, public parameters para, and sk. It outputs a private key d_{U}.
- Enc(m, U$'$, para): The encryption algorithm takes as input a message m, attribute set U$'$, and public parameters para. It outputs ciphertext $\mathcal{C}$.
- Dec($\mathcal{C}$, U$'$, para, U, d_{U}): The decryption algorithm takes as input a ciphertext $\mathcal{C}$ for U$'$, public parameters para, and secret key d_{U} with respect to U. It first checks whether the number of attributes in U that cover the attributes from U$'$ is at least d. If it is true, output the plaintext m with d_{U}. Otherwise, output a symbol of $\perp$.

B. Security Model

Because the HABE can be viewed as a generalization of ordinary ABE, the security requirements for HABE is also indistinguishable against adaptively chosen attributes and chosen ciphertext attacks (IND-Atr-CCA). Description of the security game is the same as ABE, except that the attributes here are hierarchical. The formal definition of IND-Atr-CCA is based on the following game involving an adversary $\mathcal{A}$.

Game IND-Atr-CCA

- **Setup**(d). The challenger chooses a sufficiently large security parameter 1^λ and runs Setup to get a key pair (pk, sk) and other public parameters para. It retains secret key sk and gives pk, para to $\mathcal{A}$.
- **Phase 1**. $\mathcal{A}$ can perform a polynomially bounded number of queries in an adaptive manner to the oracles, includ-

ing attribute private key extraction oracle and ciphertext decryption oracle.

- **Challenge**. $\mathcal{A}$ outputs a target attribute set U^* and two messages m_0, m_1 on which it wishes to be challenged. The only restriction is that $\mathcal{A}$ did not previously issue a key query on U such that the number of attributes in U that cover the attributes in U^* is not less than d. The challenger randomly chooses a bit $b \in \{0,1\}$, computes $\mathcal{C} = \mathsf{Enc}(m_b, \mathsf{U}^*, \mathsf{para})$ and sends $\mathcal{C}$ to $\mathcal{A}$.
- **Phase 2**. $\mathcal{A}$ can perform a polynomially bounded number of queries to the decryption and private key extraction oracles in an adaptive manner. $\mathcal{A}$ is not allowed to issue decryption query on $(\mathcal{C}, \mathsf{U})$ or private key query on an attribute set U such that the number of attributes in U that cover the attributes in U^* is not less than d.
- **Guess**. $\mathcal{A}$ outputs a guess bit b'.

$\mathcal{A}$ wins the game if $b = b'$. The advantage of $\mathcal{A}$ in game IND-Atr-CCA is defined as the probability that $\mathcal{A}$ wins the game minus $1/2$.

In this paper, we also use a weaker notion called indistinguishable against selective attributes and chosen plaintext attacks (IND-sAtr-CPA). The definition is similar to IND-Atr-CCA, except here it requires the adversary to submit its challenge target attribute set U^* before the setup phase. Furthermore, according to the definition of chosen plaintext attack, the decryption oracle is not available to the adversary. Also, the attributes in the challenge ciphertext should be chosen in different hierarchy components. Actually, the selective model has been used in many other papers to get hierarchical identity-based encryption [4]. However, it is still an open problem to construct efficient and fully secure schemes without the selective secure model in hierarchical identity-based encryption.

C. HABE Scheme with Tree hierarchy

In this construction, the attributes are assumed to be divided into n trees with roots $\omega_{10}, \cdots, \omega_{n0}$. For the tree with root ω_{i0}, we assume its depth is ℓ_i. Let ω_{ik} be an attribute of depth k with path $(\omega_{i0}, \cdots, \omega_{ik})$ from root ω_{i0}. It is easy to verify that this construction is indeed a generalization of ABE. When all attributes are independent, *i.e.*, they do not have any relationship for access control, the construction is just an ordinary ABE. Similar to other constructions of ABE, the number d, which will be used as the distance for the decryption, should be chosen and defined in setup algorithm.

We now give a brief review on the property of pairings and its related hard problems that will be used in this paper. 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 $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a map with the following properties: 1). 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 $a, b \in_R \mathbb{Z}_p$; 2). Non-degeneracy: there exists $g_1, g_2 \in \mathbb{G}_1$ 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 $\mathbb{G}_2$; 3). Computability: There is an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}_1$. Throughout this paper, we assume that there is a trusted setup algorithm that takes as input a security parameter 1^λ and outputs the setup (p, $\mathbb{G}_1$, $\mathbb{G}_2$, g, $\hat{e}$), where group $\mathbb{G}_1 =< g >$ of prime order p has a bilinear map $\hat{e}$, and $\hat{e}(g,g)$ generates $\mathbb{G}_2$ (which also has order p). We also define the Lagrange coefficient $\Delta_{i,S}$ for $i \in Z_p$ and a set S with elements in Z_p:

$$\Delta_{i,S} = \prod_{\eta \in S, \eta \neq i} \frac{x - \eta}{i - \eta}$$

Setup(d): Let $\mathbb{G}_1$ be the bilinear group of prime order p and g be a generator of $\mathbb{G}_1$. Additionally, let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a bilinear map. Assume there are N attributes in universe and n trees are formed based on the relationship of these attributes defined in the access control system. Define a hash function $H : \{0,1\}^* \to Z_p^*$. Let U_0=$\{\omega_{10}, \cdots, \omega_{n0}\}$ be the root attributes set. Assume the maximum depth of the i-th tree is ℓ_i for $1 \leq i \leq n$, and $\ell = \max\{\ell_1, \cdots, \ell_n\}$. We can choose α from $\mathbb{Z}_p$ and compute $g_1 = g^\alpha$. Meanwhile, we choose random elements g_2, $u_1', \cdots, u_n'$, $u_1, \cdots$, u_ℓ from group $\mathbb{G}_1$.

The public parameters are $\mathsf{para} = (g, g_1, g_2,\ \hat{e},\ (u_i')_{1 \leq i \leq n}$, $(u_i)_{1 \leq i \leq \ell})$. The master key is α.

KeyGen: To generate a private key for attribute set U, it proceeds as follows:

- A $d-1$ degree polynomial q is randomly chosen such that $q(0) = \alpha$;
- For each $\omega \in \mathsf{U}$, assume its depth is k in the i-th tree with path $(\omega_{i0}, \omega_{i1}, \cdots, \omega_{i,k-1}, \omega)$. It chooses $r \in_R \mathbb{Z}_p$ and computes D_ω=(d_{i0}, d_i, d_{i,k_i+1}, $\cdots$, $d_{i\ell_i}$), where $d_{i0} = g_2^{q(H(\omega))}(u_i' \Pi_{j=1}^k u_j^{\omega_{ij}})^r$, $d_i = g^r$, $d_{i,k+1}$=u_{k+1}^r, $\cdots$, $d_{i\ell_i}$=$u_{\ell_i}^r$;
- Finally, it outputs the private key of U as $d_{\mathsf{U}} = \{D_\omega\}_{\omega \in \mathsf{U}}$.

Enc: To encrypt a message $m \in \mathbb{G}_2$ to an attribute set U', it proceeds as follows. First, a random value $s \in \mathbb{Z}_p$ is chosen. For each $\omega' \in \mathsf{U}'$, assume its depth is k' in the j-th tree. Let the path for ω' be $(\omega_{j0}, \omega_{j1}',\ \cdots,\ \omega_{j,k'-1}',\ \omega')$. It computes $E' = m\hat{e}(g_1, g_2)^s$ and $T = g^s$. Furthermore, it computes $E_{\omega'} = (u_j' \prod_{\delta=1}^{k'} u_\delta^{\omega_{j\delta}'})^s$ for each $\omega' \in \mathsf{U}'$ and outputs the ciphertext as $\mathcal{C}$=(E', T, $\{E_{\omega'}\}$) for all $\omega' \in \mathsf{U}'$.

Dec: Suppose that a ciphertext E is encrypted to the attribute set U'. Assume one has a private key $d_{\mathsf{U}} = \{D_\omega\}_{\omega \in \mathsf{U}}$ for attribute set U such that the number of attributes in U that cover the attributes in U' is no less than d. Then, it chooses an arbitrary d-element subset S with elements in U. For each ω in S with path $(\omega_{i0}, \omega_{i1}, \cdots, \omega_{i,k-1}, \omega)$, assume ω' is the attribute in U' covered by ω with path from the same root ω_{i0} as $(\omega_{i0}, \omega_{i1}',\ \cdots,\ \omega_{i,k'-1}',\ \omega')$ (It implies the depth for ω' is k' in the j-th tree). Then, we have $\omega_{i\delta} = \omega_{i\delta}'$ for $1 \leq \delta \leq k$. Finally, it computes $d_{i0}' = d_{i0} d_{i,k+1}^{\omega_{i,k+1}'} \cdots d_{ik'}^{\omega_{ik'}'}$ and decrypts the ciphertext as

$$m = E' / \prod_{\omega \in S} \left(\frac{\hat{e}(d_{i0}', T)}{\hat{e}(d_i, E_{\omega'})}\right)^{\triangle_{H(\omega),S}(0)}$$

3.3.1 Security Result

Before giving the security result, we introduce the Decisional ℓ-wBDHI* Assumption used in [2].

DECISIONAL ℓ-wBDHI* PROBLEM: The Decisional ℓ-wBDHI* Problem is that, given g, $y_1 = g^x, \cdots, y_\ell = g^{x^\ell} \in \mathbb{G}_1$ for unknown random $x \in \mathbb{Z}_p^*$ and $T \in \mathbb{G}_2$, to decide if $T = \hat{e}(g,g)^{x^{\ell+1}}$.

We say that a polynomial-time adversary $\mathcal{A}$ has advantage ϵ in solving the Decisional ℓ-wBDHI* Problem in groups $(\mathbb{G}_1, \mathbb{G}_2)$ if $|\ Pr[\mathcal{A}(g,\ y_1 = g^x, \cdots, y_\ell = g^{x^\ell},\ \hat{e}(g,g)^{x^{\ell+1}}) = 1]$- Pr $[\mathcal{A}(g,\ y_1 = g^x, \cdots, y_\ell = g^{x^\ell},\ \hat{e}(g,g)^z) = 1]\ | \geq 2\epsilon$, where the probability is taken over the randomly chosen x, z and the random bits consumed by $\mathcal{A}$.

DECISIONAL ℓ-wBDHI* ASSUMPTION: We say that the (t,ϵ)-ℓ-wBDHI* assumption holds in $(\mathbb{G}_1, \mathbb{G}_2)$ if no t-time algorithm has the probability at least ϵ in solving the ℓ-wBDHI* problem for non-negligible ϵ.

***Theorem** 1:* Under the ℓ-wBDHI* assumption, the HABE scheme is indistinguishable secure against selective-attribute chosen plaintext attack.

Proof: See Appendix A. ■

Similar to [23], we have two methods to obtain flexible d. First, we can create multiple systems with different values of d and one can encrypt message by choosing the appropriate value d. In the second method, the attribute authority will reserve some root attributes which will be issued to everyone. The party encrypting the message can decrease d by increasing the number of these '*default*' attributes included in the ciphertext.

The most efficient transformation from IND-sAtr-CPA to IND-sAtr-CCA is to use the Fujisaki-Okamoto technique [11], which adds only a little computation overhead on the original HABE scheme. Thus, the resulted IND-sAtr-CCA HABE construction is very efficient.

III. Implementation and Efficiency Analysis

In the HABE with tree hierarchy, the attributes are first classified according to the relationships defined in the access control system. Assume there are n trees formed by part of universal attributes, and the remaining attributes are independent as the ordinary ABE. Actually, the independent attributes can be also viewed as trees with only roots, which is a special case from our HABE construction. Each attribute belongs to only one different tree. In HABE, the private key of higher level attributes can be utilized to decrypt the ciphertext for lower attributes. Similar to other ordinary ABE schemes, the encryptor defines the attributes set included in the ciphertext. The users are issued private keys of some attributes by the attribute center. If the user has several attributes belonging to the same path, then, only the highest level attribute will be issued. This is because in this access control system, the highest level attribute will cover all of its decedents in decryption. In ciphertext, the case is opposite. If there are several attributes belonging to the same path, only the lowest attribute will be included to create the ciphertext. This is because if one user has a private key for a higher level attribute, he or she can definitely decrypt the ciphertext for the lower level attributes. From the private key issuing, we can also understand the rule of this ciphertext generation. In decryption algorithm, only users with at least d of attributes that cover the attributes in ciphertext can decrypt the ciphertext. In our construction, the ciphertext consists of only $2+k$ group elements, where k is the size of user's attributes. If we directly apply the ABE here to realize the attribute hierarchical structure, $2 + k + \Sigma_{i=1}^{k} N_i$ group elements will be required in the ciphertext, where N_i is the number of the i-th target attribute's ancestors. There is also another way to reduce the ciphertext size by just issuing keys with all decedents of the user's attributes. However, the attribute private key size will be $2(k + N_i')$, where N_i' is the number of the i-th target decedents.

IV. Conclusion and Future Work

ABE has been applied extensively to the area of access control. However, the application of ABE is limited due to its high scheme overhead as extensive pairing operations are usually required.

In this paper, we focus on improving the efficiency of ABE by leveraging a previously overlooked fact, i.e., the often-found hierarchical relationships among the attributes that are inherent to many access control scenarios. As the first research effort along this direction, we coin the notion of hierarchical ABE (HABE), which can be viewed as the generalization of traditional ABE in the sense that both definitions are equal when all attributes are independent. We further give a concrete HABE construction considering a tree hierarchy among the attributes, which is provably secure. More importantly, our construction can exhibit significant improvement over the traditional ABE when attribute hierarchies exist.

This paper is the first work to address how to improve ABE by considering the relationships among the attributes. There are still several interesting open problems in this topic: 1) How can we construct more efficient schemes with attribute tree hierarchical structure? 2) How can we improve ABE by designing constructions dealing with more general relationships among the attributes in universe? In this paper, we consider the most common attributes structure, *i.e.*, tree structure. Other attributes structure, such as partial-order tree, can also be utilized in some scenarios. Therefore, how to design ABE for more general attributes structure is our future work.

Acknowledgement

This work was supported in part by the US National Science Foundation under grant CNS-0831963.

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] D. Boneh, X. Boyen and E.Goh. *Hierarchical Identity based encryption with constant ciphertext*. EUROCRYPT'05. LNCS 3494, pp. 440-456, springer, 2005.

[3] John Bethencourt, Amit Sahai, and Brent Waters. *Ciphertext-Policy Attribute-Based Encryption*. IEEE Symposium on Security and Privacy'07, pp. 321-334, IEEE, 2007.

[4] Dan Boneh and Xavier Boyen. *Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles*. EUROCRYPT'04, LNCS 3027, pp. 223-238, Springer, 2004.

[5] Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. *Chosen-Ciphertext Security from Identity-Based Encryption*. SIAM J. Comput. 36(5): 1301-1328, 2007.

[6] Dan Boneh and M. Franklin. *Identity-Based Encryption from The Weil Pairing*, Crypto'01, LNCS 2139, pp.213-229, Springer-Verlag, 2001.

[7] Dan Boneh and Brent Waters. *Conjunctive, Subset, and Range Queries on Encrypted Data*. TCC'07. LNCS 4392, pp. 535-554. Springer, 2007.

[8] 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.

[9] Melissa Chase. *Multi-Authority Attribute Based Encryption*. TCC'07, LNCS 4392, pp. 515-534, Springer, 2007.

[10] 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.

[11] Eiichiro Fujisaki and Tatsuaki Okamoto. *Secure Integration of Asymmetric and Symmetric Encryption Schemes*.CRYPTO'99, LNCS 1666, pp. 537-554, Springer, 1999.

[12] 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.

[13] 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.

[14] Vipul Goyal, Abhishek Jain, Omkant Pandey, and Amit Sahai. *Bounded Ciphertext Policy Attribute Based Encryption*. ICALP'08. LNCS 5126, pp. 579-591, 2008.

[15] Vipul Goyal, Steve Lu, Amit Sahai, and Brent Waters. *Black-Box Accountable Authority Identity-Based Encryption*. CCS'08, USA.

[16] 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.

[17] 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.

[18] 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.

[19] J. Li and Kwangjo Kim. *Attribute-Based Ring Signature*. Available at http://eprint.iacr.org/2008/394, 2008.

[20] H. Maji, M. Prabhakaran, and M. Rosulek. *Attribute Based Signatures: Achieving Attribute Privacy and Collusion-Resistance*, Available at http://eprint.iacr.org/2008/328, 2008.

[21] Takashi Nishide, Kazuki Yoneyama, and Kazuo Ohta. *ABE with Partially Hidden Encryptor-Specified Access Structure*. ACNS'08, LNCS 5037, pp. 111-129, Springer, 2008.

[22] Rafail Ostrovsky, Amit Sahai, and Brent Waters. *Attribute-based Encryption with Non-Monotonic Access Structures*. CCS'07, pp. 195-203, ACM, 2007.

[23] Amit Sahai and Brent Waters. *Fuzzy Identity-Based Encryption*. EUROCRYPT'05, LNCS 3494, pp. 457-473, Springer, 2005.

[24] Ravinderpal S. Sandhu. *Cryptographic Implementation of a Tree Hierarchy for Access Control*. Information Processing Letters, Volume 27, Issue 2, pp. 95-98, 1988.

[25] Adi Shamir. *How to Share a Secret*. vol. 22, pp. 612-613, ACM, 1979.

[26] Adi Shamir. *Identity-Based Cryptosystems and Signature Schemes*. CRYPTO'84, LNCS 196, pp. 47-53, Springer, 1984.

[27] Brent Waters. *Efficient Identity-Based Encryption Without Random Oracles*. EUROCRYPT'05, LNCS 3494, pp. 114-127, Springer, 2005.

APPENDIX A: PROOF OF THEOREM 1

Proof: Assume that an attacker $\mathcal{A}$ breaks IND-sAtr-CPA with probability greater than ϵ within time t by making q_d private key extraction queries. Assume the attributes in universe form n trees. Denote depth of the i-th tree as ℓ_i for $1 \leq i \leq n$, and let $\ell = \max\{\ell_1, \cdots, \ell_n\}$. We show that using $\mathcal{A}$, one can construct a ℓ-wBDHI* attacker $\mathcal{A}'$. Let g be a generator of $\mathbb{G}_1$ and $y_i = g^{x^i}$. Suppose that $\mathcal{A}'$ is given $(g, \hat{e}, \mathbb{G}_1, \mathbb{G}_2, h, y_1, \cdots, y_\ell, T)$, where T is either $\hat{e}(g,g)^{x^{\ell+1}}$ or $\hat{e}(g,g)^\gamma$ for random $\gamma \in \mathbb{Z}_p$, as an instance of the ℓ-wBDHI* problem. By ϵ' and t', we denote winning probability and running time of $\mathcal{A}'$, respectively. $\mathcal{A}$ is first given the attributes relationship trees for the access control system in advance. Then, algorithm $\mathcal{A}'$ works by interacting with $\mathcal{A}$ in a selective identity game as follows:

Suppose that $\mathcal{A}$ outputs challenge attributes U^*. Let $\mid \mathsf{U}^* \mid = \upsilon$ and $\mathsf{U}^* = (\omega^*_{i_1}, \cdots, \omega^*_{i_\upsilon})$ with the depth $k_1, \cdots, k_\upsilon$, respectively. The path for ω^* is defined as $(\omega^*_{i0}, \cdots, \omega^*_{i,k_i-1}, \omega^*)$ with depth k_i from the root ω^*_{i0} in the i-th tree. Upon receiving the challenge attributes, $\mathcal{A}'$ sets $g_1 = y_1$, $g_2 = y_\ell$, and $u_i = y_{\ell-i+1}$ for $1 \leq i \leq \ell$. For any $i \notin \{i_1, \cdots, i_\upsilon\}$, it chooses a_i from $\mathbb{Z}_p^*$ and set $u'_i = g^{a_i}$. For $i \in \{i_1, \cdots, i_\upsilon\}$, let $u'_i = g^{a_i}/\Pi_{\delta=1}^{k_i} y_{\ell-i+1}^{\omega^*_{i\delta}}$. para=$(g, e, \mathbb{G}_1, \mathbb{G}_2, g_1, g_2, d, (u'_i)_{1\leq i\leq n}, (u_i)_{1\leq i\leq \ell})$ is given to $\mathcal{A}$.

$\mathcal{A}'$ answers $\mathcal{A}$'s attributes private key extraction queries as follows. Upon receiving a private key extraction query on U, it constructs an attributes subset Γ from U such that the attributes in Γ cover attributes in U^*. We also define Γ' such that $\Gamma \subseteq \Gamma' \subseteq \mathsf{U}$ and $\mid \Gamma' \mid = d-1$. Let $S = \Gamma' \cup \{0\}$. For each $\omega \in \Gamma'$, a random value μ is chosen and let $q(H(\omega)) = \mu$.

Then, the $d-1$ degree polynomial function $q(z)$ could be determined from these $d-1$ value together with $q(0) = x$. By using interpolation, for $\omega \notin S$, $q(H(\omega))=\Sigma_{\omega\in\Gamma'}\Delta_{\omega,S}(H(\omega))q(H(\omega))+\Delta_{0,S}(H(\omega))q(0)$. So, the simulator can calculate the private key for $\omega \in S$ as $D_\omega=(d_{i0}, d_i, d_{i,k_i+1}, \cdots, d_{i\ell_i})$, where $d_{i0} = g_2^{q(H(\omega))}(u'_i\Pi_{j=1}^{k_i} u_j^{\omega_{ij}})^r$, $d_i = g^r$, $d_{i,k_i+1}=u^r_{k_i+1}, \cdots, d_{i\ell_i}=u^r_{\ell_i}$ by choosing randomly $r \in Z_p^*$. Thus, the simulator can calculate the private key D_ω for $\omega \notin S$ as follows: For $(\omega_{j0}, \omega_{j1}, \cdots, \omega_{jt_j})$, if $j \in \{i_1, \cdots, i_\ell\}$, there is at least one $1 \leq \gamma \leq t_j$, such that $\omega_{j\gamma} \neq \omega^*_{j\gamma}$. It chooses $r_j = \frac{-\Delta_{0,S}(j)x}{a_j} + r'_j$ and outputs the simulated private key as $(g_2^{\Sigma_{i\in\Gamma'}\Delta_{i,S}(j)q(j)+\frac{-\Delta_{0,S}(j)b_{j\gamma}\omega^*_{j\gamma}\omega_{j\gamma}}{\omega^*_{j\gamma}-\omega_{j\gamma}}} g_1^{(1-\frac{\omega_{j\gamma}}{\omega^*_{j\gamma}})r'_j} g^{b_{j\gamma}\omega_{j\gamma}r_j} \prod_{k\neq\gamma,k=1}^{k_j}(g_1 u_{jk})^{r'_j}, g_2^{\frac{-\Delta_{0,S}(j)\omega^*_{j\gamma}}{\omega^*_{j\gamma}-\omega_{j\gamma}}} g^{r'_j}, u^r_{k_j+1}, \cdots, u^r_{\ell_j})$.

If $j \notin \{i_1, \cdots, i_\ell\}$, then let $r_j = \frac{-\Delta_{0,S}(j)x}{a_j} + r'_j$. Finally, it outputs the simulated private key as $(g_2^{\Sigma_{j\in\Gamma'}\Delta_{j,S}(i)q(j)} g_2^{\frac{-\Delta_{0,S}(j)}{a_j}} u_j^{r'_j} \prod_{\delta=1}^{k_j}(g_1 u_{j\delta})^{r_{j\delta}}, g_2^{\frac{-\Delta_{0,S}(j)}{a_j}} g^{r'_j}, u^r_{k_j+1}, \cdots, u^r_{\ell_j})$.

After these interactions, $\mathcal{A}$ outputs two messages m_0, m_1 and U^*. $\mathcal{A}'$ picks a random bit $b \in \{0,1\}$ and responds with the ciphertext as $\mathcal{C} = (Tm_b, y_1, \{y_1^{a_j}\}_{1\leq j\leq \upsilon})$. The ciphertext is simulated correctly if $T = e(g,g)^{x^{\ell+1}}$ because let $s = x$, the ciphertext could be written as $\mathcal{C} = (m\hat{e}(g_1,g_2)^s, g^s, \{(u'_j\prod_{\delta=1}^{k_\delta} u_\delta^{\omega^*_j})^s\})$ for each $\omega \in \mathsf{U}$. $\mathcal{A}$ issues more private key queries U, restriction is that U is not covered by U^*. $\mathcal{A}'$ responds as before.

This completes the description of algorithm $\mathcal{A}'$. Finally, $\mathcal{A}$ outputs guess b' with advantage ϵ'. If $\mathcal{A}'$ does not abort, then, $\mathcal{A}'$ outputs b' as the result to the ℓ-wBDHI* problem. Since $\mathcal{A}$ has an advantage ϵ in attacking the scheme, from the simulation, we can infer that $\mathcal{A}'$ can solve the ℓ-wBDHI* problem with advantage $\epsilon' \approx \epsilon$.

■