Comments and Corrections

Comments on "Control Cloud Data Access Privilege and Anonymity With Fully Anonymous Attribute-Based Encryption"

Hui Ma, Rui Zhang, and Wei Yuan

***Abstract*—Most of the known attribute-based encryption (ABE) schemes focused on the data contents privacy and the access control, but less attention was paid to the privilege control and the identity privacy problem. Recently in IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (TIFS) (DOI:10.1109/TIFS.2014.2368352), Jung *et al.* proposed an anonymous attribute-based encryption scheme for access privilege and anonymity, which exhibited a lot of interesting ideas and gave the proof in the standard model. However, after carefully revisiting the scheme, we found that any valid user can compute the system-wide master key and their proof has some mistakes, hence, it fails to meet their security definitions.**

***Index Terms*—Cryptanalysis, attribute-based encryption, anonymity, access privilege.**

I. INTRODUCTION

As attribute-based encryption (ABE) can provide fine-grained (non-interactive) access control and encryption functionalities simultaneously, it has become a promising technique for cloud computing. Recently in the above paper [1], Jung et al. proposed an innovative ABE scheme for semi-anonymity and access privilege in the standard model, based on which, they decentralized the central authority to limit the identity leakage and introduced the file privilege control to manage all the operations on the cloud data. Furthermore, they gave the proof under the Decisional Bilinear Diffie-Hellman (DBDH) assumption. However, after carefully analyzing their scheme, we found that any valid user can compute the system-wide master key, thus is able to decrypt any ciphertext in the system. Also there are some mistakes in their security proofs.

In particular, Jung et al. introduced N authorities to decentralize the central authority, namely, each authority controls a subset of attributes. The authors of [1] also extended the original access tree to several privilege trees, where each privilege tree described one operation on the cloud data. Due to the absence of a central authority, all the authorities should work jointly to create the master key for each authority, the private key for each user, the public parameter and the master key for the whole system. In a closer view, based on [2], the authors introduced a novel technique that is claimed to resist collusion of at most N-2 authorities. However, as shown later, after receiving a valid private key, a user can easily recover the system-wide master key, then he can generate any secret key for any attribute and decrypt any ciphertext in the system. In other words, the scheme in [1] is insecure.

II. PRELIMINARIES

A. Notations

Let $\mathcal{A}_k$, $\mathbf{A}$ and $\mathbb{A}^u$ denote the k-th authority, all the authorities and the user u's attribute set, respectively. T_l denotes a tree that describes a privilege l. $\mathbb{A}^{T_l}$ denotes the attribute set included in tree T_l and $\{T_l\}_{l \in \{0,\ldots,r\text{-}1\}}$ denotes a set of r privilege trees, respectively.

B. Review of the Scheme

We briefly review the semi-anonymous ABE scheme [1] below.

- $\mathbf{Setup} \rightarrow (\mathbf{PK}, \mathbf{MK_k})$. One of N authorities chooses a bilinear group $\mathbb{G}_0$ of prime order p with generator g and publishes it. All the authorities randomly pick $v_k \in \mathbb{Z}_p$ and send $Y_k = e(g,g)^{v_k}$ to others who individually compute $Y = \prod_{k \in \mathbf{A}} Y_k = e(g,g)^{\sum_{k \in \mathbf{A}} v_k}$. Each authority $\mathcal{A}_k$ randomly picks N-1 integers $s_{kj} \in \mathbb{Z}_p (j \in \{1, \ldots, N\} \backslash \{k\})$, computes $g^{s_{kj}}$ that is shared with other authority $\mathcal{A}_j$. An authority $\mathcal{A}_k$, after receiving N-1 pieces of $g^{s_{jk}}$, computes $x_k \in \mathbb{Z}_p$:
$$x_k = (\prod_{j \in \{1,\ldots,N\} \backslash \{k\}} g^{s_{kj}}) / (\prod_{j \in \{1,\ldots,N\} \backslash \{k\}} g^{s_{jk}}) = g^{\left(\sum_{j \in \{1,\ldots,N\} \backslash \{k\}} s_{kj} - \sum_{j \in \{1,\ldots,N\} \backslash \{k\}} s_{jk}\right)}.$$
The master key for $\mathcal{A}_k$ is $MK_k = \{v_k, x_k\}$, and public key of the whole system is $PK = \{\mathbb{G}_0, g, Y\}$.
- $\mathbf{KeyGen}(\mathbf{PK}, \mathbf{MK_k}, \mathbb{A}^{\mathbf{u}}) \rightarrow \mathbf{SK_u}$. A new user u should request the private key from all of the authorities as follows:
  1) *Attribute Key Generation:* For any attribute $i \in \mathbb{A}^u$, every $\mathcal{A}_k$ randomly picks $r_i \in \mathbb{Z}_p$ to compute $H(att(i))^{r_i}$, $D'_i = g^{r_i}$, which are sent to the user. Then each authority $\mathcal{A}_k$ randomly picks $d_k \in \mathbb{Z}_p$, computes $x_k \cdot g^{v_k} \cdot g^{d_k}$ and privately shares it with other authorities (i.e. kept secret to the user). $\mathcal{A}_k$ privately sends $x_k \cdot g^{d_k}$ to the user (i.e. kept secret to other authorities). Next, one of N authorities computes $D = \prod x_k g^{v_k} g^{d_k} = g^{\sum v_k + \sum d_k}$ and sends it to the user. At last, the user computes $D_i = H(att(i))^{r_i} \cdot \prod (x_k \cdot g^{d_k}) = H(att(i))^{r_i} \cdot g^{\sum d_k}$.
  2) *Key Aggregation:* After receiving D, D_i and D'_i, the user aggregates the components as his private key: $SK_u = \{D, \forall i \in \mathbb{A}^u : D_i = H(att(i))^{r_i} \cdot g^{\sum d_k}, D'_i = g^{r_i}\}$.
- $\mathbf{Encrypt}(\mathbf{PK}, \mathbf{M}, \{\mathbf{T}_l\}_{l \in \{0,\ldots,r\text{-}1\}}) \rightarrow (\mathbf{CT}, \mathbf{VR})$. For each T_l, the algorithm acts as the scheme in [2] except that the encrypted message is K_e that is the decryption key of the symmetric encryption scheme; the algorithm sets $q_{R_l}(0) = s_l$, picks

Manuscript received March 31, 2015; revised August 9, 2015; accepted August 19, 2015. Date of publication December 18, 2015; date of current version February 1, 2016. This work was supported in part by the Foundation of Science and Technology on Information Assurance Laboratory under Grant KJ-14-002, in part by the Strategic Priority Research Program through the Chinese Academy of Sciences (CAS) under Grant XDA06010703, in part by the National Natural Science Foundation of China under Grant 61402468, and in part by the One Hundred Talents Project of CAS. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Liqun Chen. *(Corresponding author: Rui Zhang.)*

H. Ma is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: mahui@iie.ac.cn).

R. Zhang and W. Yuan are with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: r-zhang@iie.ac.cn; yuanwei@iie.ac.cn).

Digital Object Identifier 10.1109/TIFS.2015.2509865

a random element $h \in \mathbb{Z}_p$ such that h^{-1} mod p exists and calculates $g^{h \cdot s_l}$, $D^{h^{-1}}$. The ciphertext **CT** is:

$$\begin{aligned}&\langle \{T_l\}_{l \in \{0,\ldots,r-1\}}, E_0 = K_e \cdot Y^{s_0}, C = g^{hs_l}, \widehat{C} = D^{h^{-1}},\\ &\quad \{C_i = g^{q_i(0)}, C_i' = H(att(i))^{q_i(0)}\}_{i \in \mathbb{A}^{T_l}, \forall l \in \{0,\ldots,r-1\}}\rangle.\end{aligned}$$

VR $= \langle \{E_l = Y^{s_l}\}_{l \in \{1,\ldots,r-1\}} \rangle$, which is disclosed only to the cloud server.

- **Decrypt**$(\mathbf{PK}, \mathbf{SK_u}, \mathbf{CT}) \rightarrow (\mathbf{M}, \mathbf{VR})$. For each T_l, the algorithm acts as the scheme in [2] except the last step. If the user wants to read the file, the decryption key K_e can be recovered by
$$\frac{E_0}{\frac{e(C,\widehat{C})}{e(g,g)^{s_0 \sum d_k}}} = \frac{K_e \cdot Y^{s_0}}{\frac{e(g,g)^{s_0(\sum d_k + \sum v_k)}}{e(g,g)^{s_0 \sum d_k}}} = K_e,$$
where $e(g,g)^{s_0 \sum d_k}$ is the decryption result of the root node of T_0. Otherwise, if the user wants to execute some operations on the data, he should decrypt the corresponding privilege tree, recover Y^{s_j} and send it to the cloud server. The cloud server checks whether $Y^{s_j} = E_j$ and proceeds if they do equal.

C. The Security Model

We briefly review the security model in [1] below.

- **Init:** The adversary $\mathcal{A}$ declares the set of compromised authorities $\{\mathcal{A}_k\} \subset \mathbf{A}$ (where at least two authorities in $\mathbf{A}$ are not controlled by $\mathcal{A}$) that are under his control ($\mathbf{A}/\{\mathcal{A}_k\}$ are controlled by the challenger $\mathcal{C}$). Then $\mathcal{A}$ declares T_0 that he wants to be challenged, in which some attributes are being in charged by $\mathcal{C}$'s authorities.
- **Setup*:** $\mathcal{C}$ and $\mathcal{A}$ jointly run the *Setup* algorithm to receive the valid outputs.
- **Phase 1:** $\mathcal{A}$ launches *KeyGen* algorithms to query for as many private keys as he wants, which correspond to attribute sets $\mathbb{A}_1, \ldots, \mathbb{A}_q$ being disjointly in charged by all authorities, but none of these keys satisfy T_0.
- **Challenge:** $\mathcal{A}$ submits two messages M_0 and M_1 of equal size to $\mathcal{C}$. $\mathcal{C}$ flips a random binary coin b and encrypts M_b with T_0. The ciphertext CT is given to $\mathcal{A}$.
- **Phase 2:** Phase 1 is repeated adaptively, but none of the queried keys satisfy T_0.
- **Guess:** $\mathcal{A}$ outputs a guess b' of b.

The advantage of $\mathcal{A}$ in this game is defined as $\Pr[b' = b] - \frac{1}{2}$.

Definition 1: The scheme is secure and indistinguishable against chosen-attribute attack (IND-CAA) if all probabilistic polynomial-time adversaries (PPTA) have at most a negligible advantage in the above game.

Note that 1) the above security game is selective, which means we add an **Init** stage before **Setup*** where the adversary commits to the challenge access structure at the beginning. 2) the IND-CAA defined above implies IND-CCA since the adversary can conduct encryptions and decryptions using the public keys and secret keys it owns in **Phase 1** and **Phase 2** (but he cannot decrypt the target ciphertext since none of its secret keys satisfy T_0).

III. Security Analysis of Jung *et al.*'s ABE Scheme

A. Security Analysis

Though Jung et al. claimed that their scheme resists authorities' collusion attack and users' collusion attack, however, the system-wide master key used to generate a valid secret key can be computed by any valid user as follows: According to the security model and scheme, in **Init**, the adversary $\mathcal{A}$ declares that he does not control any authorities. In **Phase 1**, $\mathcal{A}$ launches *KeyGen* algorithm to query only one private key that does not satisfy T_0. After the first step in *KeyGen*, $\mathcal{A}$ receives

$$\begin{aligned}&D = g^{\sum v_k + \sum d_k} && \text{(from one of the } N \text{ authorities), and}\\ &\{x_k \cdot g^{d_k}\}_{k \in \mathcal{A}} && \text{(from every authority } \mathcal{A}_k\text{).}\end{aligned}$$

Then $\mathcal{A}$ computes

$$L = \prod (x_k \cdot g^{d_k}) = g^{\sum d_k}$$
$$D/L = \frac{g^{\sum v_k + \sum d_k}}{g^{\sum d_k}} = g^{\sum v_k},$$

where $g^{\sum v_k}$ acts as a system-wide master key used to generate a valid secret key (as was also pointed out in [1]). Once the system-wide master key is disclosed, there is no secrecy in the ABE system. Since $\mathcal{A}$ is aware of T_0, $\mathcal{A}$ can generate a private key SK^* with $g^{\sum v_k}$ and an attribute set that satisfies T_0, decrypt the challenge ciphertext and output the correct guess with probability 1.

Apparently, once $\mathcal{A}$ knows the system-wide master key, $\mathcal{A}$ can generate any private key with any attribute set and decrypt all the ciphertexts in the system.

B. Mistake in the Proofs

The proof given in [1] shows a reduction from the security of the scheme to hardness of the DBDH problem. However, we found a serious mistake in it.

In their proof, the DBDH challenger flips a binary coin u, and sets $(g, A, B, C, Z) = (g, g^a, g^b, g^c, e(g,g)^{abc})$ if $u = 0$; otherwise he sets $(g, A, B, C, Z) = (g, g^a, g^b, g^c, e(g,g)^z)$, where $a, b, c, z \in \mathbb{Z}_p$ are randomly selected. The challenger then gives the simulator (*Sim*) $\langle g, A, B, C, Z \rangle = \langle g, g^a, g^b, g^c, Z \rangle$. *Sim* plays the role of a challenger in the ABE security model and sets $a = \sum d_k, b = \frac{\sum v_k}{\sum d_k}, c = s_0$, where $d_1, \ldots, d_n, v_1, \ldots, v_n, s_0 \in \mathbb{Z}_p$ are randomly chosen. Meanwhile, *Sim* sets the parameter $Y = e(A, B) = e(g,g)^{ab}$. During the *Challenge* phase, *Sim* sets $Y^{s_0} = Z$, if $u = 0$, $Z = e(g,g)^{abc}$; otherwise, $Z = e(g,g)^z$. However, an adversary can recover $g^{ab} = g^{\sum v_k}$ in a similar way as discussed in our attack. Thus it can compute $e(g^{ab}, g^c) = e(g,g)^{abc}$ easily and solve the DBDH problem. Additionally, it is strange that in [1], *Sim* is only required to compute $D_i = A \cdot H(att(i))^{r_i}, D_i' = g^{r_i}$, but how to generate $D = g^{\sum v_k + \sum d_k}$ and $\{x_k \cdot g^{\sum d_k}\}_{k \in \mathcal{A}}$ (in a *KeyGen* query) was not mentioned.

IV. Conclusion

We analyzed the security of an ABE scheme [1] for access privilege and anonymity and showed that any valid user can compute the system-wide master secret key thus decrypt all the ciphertext in the system. Therefore, we remark that the problem of building an anonymous ABE for access privilege and anonymity is still open.

Acknowledgment

The authors thank anonymous reviewers for their valuable comments and all the authors of [1] for their positive feedback. Moreover, all the authors of [1] propose a fixed version, in which the user should receive $(x_k * g^{d_k})$'s combined with $(H(att(i))^{r_i})$'s instead of $(x_k * g^{d_k})$'s and $(H(att(i))^{r_i})$'s, respectively. In that case, the attack given in this draft does not work.

References

[1] T. Jung, X.-Y. Li, Z. Wan, and M. Wan, "Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 190–199, Jan. 2015. [Online]. Available: http://dx.doi.org/10.1109/TIFS.2014.2368352

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2007, pp. 321–334.