SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2013; **6**:1143–1152
Published online 15 February 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.683

RESEARCH ARTICLE

Free global ID against collusion attack on multi-authority attribute-based encryption

Ang Gao* and Zengzhi Li

School of Electronic and Information Engineering, Xian Jiaotong University, Xi'an, Shaanxi, China

ABSTRACT

In order to resolve the problem that collusion attack on attribute-based encryption (ABE) with multi-authority, we firstly formulize user action of making request for key into legality and collusion by the relationship between user's attributes and decryption threshold. Furthermore, we propose an ABE scheme without presenting user's global ID (GID). In our first system, a trusted central authority assists each attribute authority (AA) to independently run a security check for users' requests, such that only legal users have power to decrypt a message. In order to prevent a malicious user from passing security check by submitting duplication request to AA, we improve the first system by our second system, where the same requests of different users are transformed into different ones associated with the subset of attributes indexes. Finally, in order to adapt this transformation to ABE, discrete Fourier transform and inverse discrete Fourier transform are used to share and recover secret key, respectively. As shown in the results of security and performance evaluation, our scheme not only improves user's privacy but also is more efficient than existing ABE schemes. Copyright © 2013 John Wiley & Sons, Ltd.

KEYWORDS

free GID; attribute-based encryption; multi-authority; collusion attack; privacy protection

***Correspondence**

Ang Gao, Institute of Computer Software and Theory, Xian Jiaotong University, No. 28 Xianning West Road, Xi'an, Shaanxi, 710049, China.
E-mail: gaoang.cs@gmail.com

1. INTRODUCTION

In public key cryptosystems, Shamir [1] first proposed identity-based cryptosystems and signature scheme, where a public key is peoples unique identity (e.g., identity card No., e-mail address, and driving license No., etc.), and the private key associated with ID is issued by a trusted private key generator (PKG) over a secret channel. Similarly, human's biometric attribute (e.g., fingerprint, face, iris, and voice, etc.) is excellent in identifying an individual, needless to remember and anti-counterfeiting, such that it serves as the public key. However, there is difference in the measurements of the same biometric, which fails to decrypt message. In order to resolve the problem, Sahai and Waters [2] proposed a fuzzy identity-based encryption (IBE) scheme, where an attribute authority (AA) (as a substitute for PKG) utilizes multi-attribute provided by user to split a system-wide master secret key into a number of pieces (called subkeys), which are then issued to a user over an open channel. Only a recipient, with at least d out of n given attributes whose Hamming distances in measurements are within a certain threshold, can decrypt message. In addition to private attribute, the attributes (e.g., favorite, institution, and title, etc.) shared between users are applied to group IBE. In this way, IBE is extended to attribute-based encryption (ABE).

Many research works [3–6] have focused on ABE. With regard to collusion attack on ABE, Chase [7] proposed a multi-authority ABE scheme, where a global ID (GID) is exclusive to identify the subkeys of different users, such that the master secret key is recovered from the subkeys bound with the same GID instead of a different GID. However, the use of a trusted central authority (CA) in that construction is contradictory to the original motivation of distributed system. And what is worse, the use of the GID exposes the user's privacy as a result that AAs can combine their information related to the GID to build a full profile with all of the user's attributes. In order to improve user's privacy in fully distributed environment, Chase and Chow [8] proposed an anonymous ABE scheme without requiring CA. However, this solution was still not ideal considering primarily two drawbacks as follows:

(1) AA issues a user the subkeys bound with his or her GID, which is anonymized by non-interactive zero-knowledge proof. Accordingly, to recover the master secret key, the user has to cancel out the anonymous values of his or her subkeys by means of a distributed pseudo random function (PRF [9]), which is constructed under the assumption of a decisional bilinear Diffie–Hellman inversion with polynomial terms'

number increasing at an exponential rate. These processes incur additional computation overhead.

(2) Each AA is disallowed to link with others to prevent multi-authority ABE from degenerating into one-authority ABE. It means that as long as a user makes request for key, AA unhesitatingly responds to the user with the subkeys, which are associated with user's GID and attributes. Although it is impossible for the user to combine the subkeys copied from his or her friends into the master secret key, invalid key distribution and failed decryption incur additional costs.

2. OUR CONTRIBUTIONS

Firstly, according to the relationship between user's attributes and the threshold defined by the key policy [2,10,11] of ABE, we formulize user action of making request for key into legality and collusion. Furthermore, we formulize user action of making request for key into legality and collusion. Furthermore, we construct our first free-GID system, where a trusted CA (which is only responsible for forwarding a resending command instead of issuing key) assists each AA to independently run security check, which is dependent on the relationship between attribute multiplicity (the number of the duplications of one attribute) and the number of participant users, such that enough subkeys are only issued to legal users instead of colluders. In order to prevent a malicious user from passing security check by sending duplication request, we improve the first system by our second system, where the same requests of different users are transformed into different ones associated with the subset of attributes indexes (defined in the set of encryptor's attributes), such that AA can distinguish the duplication requests of one user from the same requests of different users. Finally, to improve security and efficiency, discrete Fourier transform (DFT [12]) and inverse discrete Fourier transform (IDFT) are used to share and recover a system-wide master secret key, respectively.

3. ATTRIBUTE-BASED ENCRYPTION PRINCIPLES

Attribute-based encryption is constructed by a bilinear map, which is denoted by $\hat{e}:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are two cyclic multiplicative groups of prime order μ generated by g_1 and g_2, respectively. For $\forall u\in\mathbb{G}_1$, $\forall v\in\mathbb{G}_2$ and $a,b\in\mathbb{Z}_\mu$, $\hat{e}$ has three properties as follows: (1) the bilinearity expressed as $\hat{e}(u^a,v^b)=\hat{e}(u,v)^{ab}$; (2) the non-degeneracy expressed as $\hat{e}(g_1,g_2)\neq 1$; (3) the computability, namely, $\hat{e}(g_1,g_2)$ is efficiently computable. Moreover, there exists an isomorphism $\psi:\mathbb{G}_2\to\mathbb{G}_1$ such that $\psi(g_2)=g_1$.

Definition 1. *Decisional Bilinear Diffie–Hellman (DBDH) problem*: *Given* $g_1\in\mathbb{G}_1$, $Z\in\mathbb{G}_T$, *and* $g_2,g_2^a,g_2^b,g_2^c\in\mathbb{G}_2$ *where* $a,b,c\in\mathbb{Z}_\mu$, *decide whether* $Z=\hat{e}(g_1,g_2)^{abc}$ *or* $Z=\hat{e}(g_1,g_2)^r$ *for random* $r\in\mathbb{Z}_\mu$.

The security of the schemes of Chase ABE and Chase–Chow ABE depends on the hardness of the DBDH problem, on which our construction also relies. To assist the reader, we list the main notations used in this paper in Table I.

4. USER ROLE

Let $\mathbb{A}^t\subseteq\mathbb{A}^e$ and $\mathbb{A}_k^t\subseteq\mathbb{A}_k^e$ denote threshold attributes set fixed by encryptor e and each AA $k\in[1,\ldots,N^c]$, respectively. The action of user $u\in\mathbb{U}$ is defined as follows:

Definition 2. *u is deemed to be legal if* $\mathbb{A}_k^u=\mathbb{A}_k^t$ *and* $\mathbb{A}^u=\mathbb{A}^t$.

Definition 3. *Let complete set be* $\mathbb{A}^t$, *u is deemed to collude with other user* $v\in\mathbb{U}/\{u\}$ *if he or she satisfies three properties as follows:* (1) $\mathbb{A}^u\subset\mathbb{A}^t$; (2) $\cup_{u\in\mathbb{U}}\mathbb{A}^u=\mathbb{A}^t$; (3) $\mathbb{A}_k^u=\overline{\cup_{v\in\mathbb{U}/\{u\}}\mathbb{A}_k^v}$.

The first property describes that each member of collusion clique has not enough attributes that decryption requires, and can be further categorized into two cases as follows: (1) $\exists\mathbb{A}_k^u\subset\mathbb{A}_k^t$, which causes multiple colluders to share one authority, and (2) $\forall\mathbb{A}_k^u\supseteq\mathbb{A}_k^t$, which causes each colluder to correspond to different authority. Regardless of which collusion is launched, it is shown in Figure 1 that multiple colluders combine their attributes with each other to achieve threshold attributes (expressed as the second property). It means that there exists a complementation relationship (expressed as the third property) between attribute sets of all colluders.

Table I. A summary of main notations.

Symbol	Meaning
$u(u_l,u_a)$, e, c	Participant user(legal user or colluder), encryptor, AA
$\mathbb{U}$	Set of users
$\mathbb{A}^e,\mathbb{A}_c,\mathbb{A}^u,\mathbb{A}_c^e,\mathbb{A}_c^u$	Attribute set of e, c, u, e corresponding to c, u corresponding to c
$\mathbb{L}^e,\mathbb{L}_c,\mathbb{L}^u,\mathbb{L}_c^e,\mathbb{L}_c^u$	Transformation set of $\mathbb{A}^e,\mathbb{A}_c,\mathbb{A}^u,\mathbb{A}_c^e,\mathbb{A}_c^u$
$\mathbb{W}^e,\mathbb{W}_c,\mathbb{W}^u,\mathbb{W}_c^e,\mathbb{W}_c^u$	Transformation set of $\mathbb{L}^e,\mathbb{L}_c,\mathbb{L}^u,\mathbb{L}_c^e,\mathbb{L}_c^u$
$p_c(\cdot)$	Polynomial of c
d_c	Degree of $p_c(\cdot)$
$m_{i,c}$	Number of the duplications of the attribute i in c
$r_{i,c}$	Resending command from c to the user with attribute i
st_u	Timestamp of u
$N^u,N^l,N^a,N^c,N_c^u,N_c^a$	Number of u, u_l, u_a, c, u corresponding to c, u_a corresponding to c

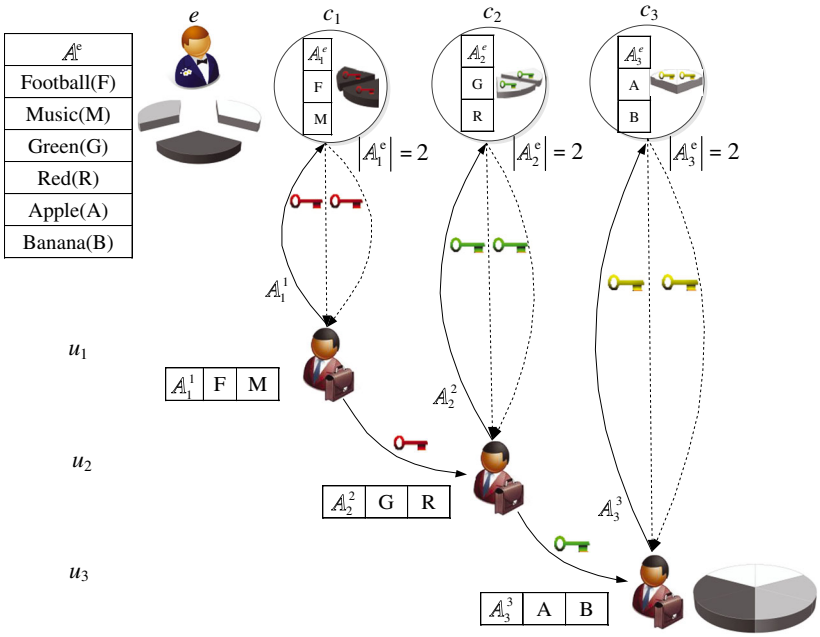


Figure 1. Collusion attack on attribute-based encryption.

There are cases other than the two definitions mentioned above. In the case that a user without enough attributes cannot exceed the decryption threshold when colluding with his or her few friends, he or she fails to decrypt sensitive message. In another case that an adversary compromises or hijacks a legal user to gain access to key, he or she does not directly participate in issuing key, which is beyond the scope of direct participation by multi-user in collusion. Therefore, we only consider that legal user and enough colluders participate in ABE.

5. OUR FIRST SYSTEM

According to Definition 3, multiple colluders can submit the union of their attribute sets to exceed threshold to AA. Unfortunately, legal user may separately submit each pieces (subsets below threshold) of his or her attribute set to AA, both of which are not distinguished by simple threshold policy. Therefore, under the assumption of unlinkable AA in Chase–Chow ABE, we require that each AA runs a security check, which is dependent on the relationship between participant users number and attribute multiplicity (the number of the duplications of one attribute).

5.1. Diagnosable attribute authority

For the convenience of denoting attribute multiplicity, we firstly define a multi-set as follows:

Definition 4. *Given a normal set* $\mathbb{S}$ *without duplication membership, a multi-set* $\mathbb{M}$ *is the set including duplication membership, and it is expressed as* $\mathbb{M} \triangleq \{m_i * i\}_{i \in \mathbb{S}}$, *where* $i \in \mathbb{S}$ *denotes different memberships in* $\mathbb{M}$ *and* m_i *denotes the number of the duplications of i, which is separated by symbol * from i.*

With the help of Definition 4, the requests from each user $j \in [1, \ldots, N_k^u]$ to each AA $k \in [1, \ldots, N^c]$ are expressed as $\{m_{i,k} * i\}_{i \in \mathbb{A}_k^e}$, and they are further categorized into three cases as follows: (1) the legal requests, which are made by only legal users; (2) the collusion requests, which are made by only colluders; (3) the hybrid requests, which are made by the mixture of legal users and colluders.

For the case of legal requests, each AA $k \in [1, \ldots, N^c]$ recognizes legal users using following theorem:

Theorem 1. *Given* $N^u \geq 1$ *participant user, the* N^u *users are legal if* $m_{i,k} = N^u$.

Proof. Assume that not all N^u users are legal, if, it is obvious that only one user is not enough to collusion if $N^u = 1$. Otherwise, there exist $n (1 < n \leq N^u)$ numbers of colluders. For $\{m'_{i,k} * i\}_{i \in \mathbb{A}_k^j}$ formed by the requests $\mathbb{A}_k^j$ of each colluder $j \in [1, \ldots, n]$, k has $m'_k = 1$ because $\mathbb{A}_k^j = \overline{\cup_{u \in \mathbb{U}/\{j\}} \mathbb{A}_k^u}$ as defined in Definition 3. For $\{m''_{i,k} * i\}_{i \in \cup_j \mathbb{A}_k^j}$ formed by the requests $\mathbb{A}_k^j$ of each legal user $j \in [1, \ldots, N^u - n]$, k has $m''_{i,k} = N^u - n$ because $\mathbb{A}_k^j = \mathbb{A}_k^e$ as defined in Definition 2. Therefore, for all N^u users' requests, k gets $m_{i,k} = m'_{i,k} + m''_{i,k} < N^u$, which is in contradiction to premise $m_{i,k} = N^u$.

For the case of collusion requests, each AA $k \in [1,\ldots,N^c]$ recognizes colluders using following theorem:

Theorem 2. *Given $N^u > 1$ participant users, all N^u users are colluders if $m_{i,k} = 1$.*

Proof. Assume that not all N^u users are colluders. Thus, participant users consist of $n(1 \le n \le N^u)$ legal user(s) and $N^u - n$ colluders. By the proof similar to that of Theorem 1, legal requests and collusion requests make AA k have $m'_{i,k} = n$ and $m''_{i,k} = 1$, respectively. Therefore, for all N^u user requests, k gets $m_{i,k} = m'_{i,k} + m''_{i,k} > 1$, which is in contradiction to premise $m_{i,k} = 1$.

In the case of hybrid requests, collusion causes $1 < m_{i,k} < N^u$, which fails AA to recognize a user by Theorem 1 and Theorem 2. Thus, we design Algorithm 1, where only the users corresponding with one AA c are required to resend their all requests until $\forall\, m_{i,k} = N^u$, such that all or most colluders are removed step by step. To prevent a user from responding to other AAs except for c in each round of Algorithm 1, a trusted CA is responsible for forwarding resending command $r_{i,c}$ of a certain AA c to all users when $r_{i,c}$ is randomly picked from the commands of all AAs. However, only the user u with $i \in \mathbb{A}^u$ responds to this command.

Moreover, AA's judgment on a user is vulnerable to be disturbed by the old requests submitted in previous round and the requests relayed by a malicious user. Thus, in Algorithm 1, we require that the user attaches a timestamp to his or her request, such that only the request, with the timestamp in a system time period *TP*, is accepted.

Finally, with the help of Algorithm 1, Theorem 1, and Theorem 2, AA is able to recognize a user in any case of requests. And the detailed steps are described in Algorithm 2.

Algorithm 1 UserExtract$\left(\{m_{i,k} * i\}_{i\in\mathbb{A}_k^j}, \{st_j\}_{j\in[1\ldots N_k^u]}, N^u\right)$

Output: n users' requests

If $\left(st_j \notin TP\right)$
  $\{m_{i,k} * i\}_{i\in\mathbb{A}_k^j} / \mathbb{A}_k^j$
Else if $(1 \le \exists m_{i,k} < N^u)$
  AA k sends $r_{i,k}$ to CA
  CA randomly picks one $r_{i,c} \in \{r_{i,k}\}_{i\in\mathbb{A}_k^j}$ and broadcasts $r_{i,c}$
  u receives $r_{i,c}$
  If $(i \notin \mathbb{A}^u)$
    u drops $r_{i,c}$
  Else
    u responds to $r_{i,c}$ by resending $\left\{\mathbb{A}_k^u\right\}_{\{k\}\subseteq[1\ldots N^c]}$
    $n \leftarrow m_{i,c} - 1 + N_c^a$
    AA k calls UserExtract$\left(\{m_{i,k} * i\}_{i\in\mathbb{A}_k^j}, n\right)$
  Else if $(\forall m_{i,k} = N^u)$
    $n \leftarrow m_{i,k}$
    Return

Algorithm 2 UserCheck$\left(\{m_{i,k} * i\}_{i\in\mathbb{A}_k^j}, \{st_j\}_{j\in[1\ldots N_k^u]}, N^u\right)$

Output: the n requests come from pure colluders (True) or not (False)

System initializes *TP*
If $\left(st_j \notin TP\right)$
  $\{m_{i,k} * i\}_{i\in\mathbb{A}_k^j} / \mathbb{A}_k^j$
Else if $\left(\mathbb{A}_k^j \bigcap \mathbb{A}_k^e = \varnothing\right)$
  Drop $\mathbb{A}_k^j$
Else
  $\mathbb{A}_k^j \leftarrow \mathbb{A}_k^j / \left(\mathbb{A}_k^j / \left(\mathbb{A}_k^j \bigcap \mathbb{A}_k^e\right)\right)$
  If $(m_{i,k} = N^u)$ **and** $(N^u \ge 1)$
    $n \leftarrow m_{i,k}$
    Return False
  Else if $(m_{i,k} = 1)$ **and** $(N^u > 1)$
    $n \leftarrow m_{i,k}$
    Return True
  Else if $(1 < m_{i,k} < N^u)$
    $n \leftarrow m_{i,k}$
    Call UserExtract$\left(\{m_{i,k} * i\}_{i\in\mathbb{A}_k^j}, n\right)$
    Return False

5.2. Construction

For each user $j \in \left[1,\ldots,N_k^u\right]$ corresponding to AA $k \in [1,\ldots,N^c]$, our ABE scheme without GID works as follows:

Setup.

- *Bilinear Group Parameter*: Let λ be a security parameter to determine the size of the group, k uses λ to generate bilinear group parameters $\langle g_1, g_2, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}(\cdot,\cdot), \psi(\cdot)\rangle$.
- *Threshold*: k fixes threshold $d_k \le |\mathbb{A}_k \cap \mathbb{A}^e|$.
- *Master Public/Private Key*: k generates $Y_k = \hat{e}(g_1, g_2)^{v_k}$ where $v_k \in \mathbb{Z}_\mu$ is picked randomly. Then, Y_k is sent to other authorities. Finally, each of them individually computes master public key $Y = \prod_{k=1}^{N^c} Y_k = \hat{e}(g_1, g_2)^{\sum_{k=1}^{N^c} v_k}$.
- *Authority Public/Private Key*: For $i \in \mathbb{A}_k^e$, k generates authority public key $T_{k,i} = g_2^{t_{k,i}}$ using a random $t_{k,i} \in \mathbb{Z}_\mu$, which serves as authority private key.

Finally, the tuple *SP* of system public parameters is published as follows:

$$SP = \left\langle d_k, Y, \left\{T_{k,i}\right\}_{i\in\mathbb{A}_k^e} \right\rangle$$

Encryption. For a message m, a random number $s \in \mathbb{Z}_\mu$ is chosen by encryptor e to publish ciphertext C as follows:

$$C = \left\langle E = mY^s, \left\{E_{k,i} = T_{k,i}^s\right\}_{i\in\mathbb{A}_k^e, \forall k\in[1,\ldots,N^c]} \right\rangle$$

Issuing Key. j submits $\mathbb{A}_k^j$ to k. Upon receipt of $\mathbb{A}_k^j$, k calls Algorithm 2 with the inputs of N^u and $\left\{m_{i,k} * i\right\}_{i\in\mathbb{A}_k^j}$.

(1) If Algorithm 2 outputs True, k rejects the n users' requests for subkeys.

(2) If Algorithm 2 outputs False, k randomly chooses a $d_k - 1$ degrees Lagrange polynomial $p_k(\cdot) \in \mathbb{Z}_\mu$. Let $v_k = p_k(0)$ be kept as a secret, k extracts the set $\left\{S_{k,i} = g_1^{p_k(i)/t_{k,i}}\right\}_{i\in\mathbb{A}_k^j}$ of subkeys, which are separately issued to the n users.

Decryption. The user j with $\left|\mathbb{A}_k^j\right| \geq d_k$ recovers m as follows:

$$m = E \Big/ \prod_{k=1}^{N^c}\prod_{n=1}^{d_k} \hat{e}\left(S_{k,i_n}, E_{k,i_n}\right)^{\lambda_k(n)}$$

where $\lambda_k(n) = \prod_{m=1,m\neq n}^{d_k} \frac{i_m}{i_m - i_n} (\mathrm{mod}\mu)$ is Lagrange interpolation coefficient.

5.3. Security

For the collusion attack on our system, we will assure that legal users have "complete" subkeys, resulting in encryption. Whereas all colluders have nothing or only the "part" of all subkeys, resulting in failure to encryption. In the cases of legal and collusion requests, it is obvious that the aim mentioned above is achieved by one round of security check. In the case of hybrid request, multiple rounds of security check are required to yield the similar result, which is stated by Theorem 3 as follows:

Theorem 3. *Given that there are $N^u = N^l + N^a$ users in making the requests for subkeys to each AA $k \in [1,\ldots,N^c]$, message m is recovered by each legal user $u_{l,n} (1 \leq n \leq N^l)$ rather than colluder $u_{a,n} (1 \leq n \leq N^a)$.*

Proof. It is shown in Figure 2 that authorities issue subkeys to users through three rounds of security check as follows:

Round 1. $N_k^u - q$ legal users and $1 \leq q < N^a$ colluders participate in key issue, and function $f : \mathbb{A}_k^e \to \left\{u_{a,n}\right\}_{n\in[1\ldots q]}$ is surjective. Thus, $\forall m_{i,k} < N^u$ results in the next round of check because

$$\left(\forall i \in \mathbb{A}_k^{u_{l,1}}\right) \wedge \cdots \wedge \left(\forall i \in \mathbb{A}_k^{u_{l,N_k^u - q}}\right)$$

and

$$\left(\forall N_k^u < N^u\right) \wedge \left(\forall q \in [1, N^a)\right)$$

Round 2. Upon receipt of $r_{i,c}$ forwarded by CA, $N_c^u - 1$ legal users and one colluder $u_a \in \left\{u_{a,n}\right\}_{n\in[1,\ldots,q]}$ corresponding to AA c are extracted to resend their all requests. Thus, $\exists\, m_{i,c} = N^u$ results from

$$\left(\exists i \in \mathbb{A}_c^{u_a}\right) \wedge \left(\forall i \in \mathbb{A}_c^{u_{l,1}}\right) \wedge \cdots \wedge \left(\forall i \in \mathbb{A}_c^{u_{l,N_c^u - 1}}\right)$$

However, for other attribute $j \in \mathbb{A}^e / \{i\}$, there exist $j \notin \mathbb{A}_k^{u_a}$ resulting in $\exists\, m_{j,k} < N^u$, which causes the next round of check.

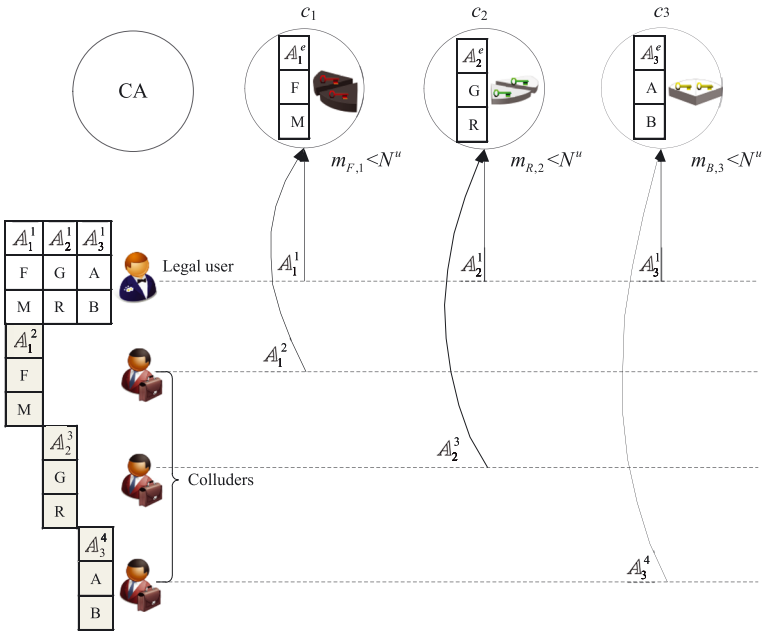
Round 3. Upon receipt of $r_{i,k}$ forwarded by CA, $N^u = N_k^u$ legal users (corresponding to AA k) rather than all colluders are extracted to resend their all requests. Thus, $\forall\, m_{i,k} = N^u$ results from

$$\left(\forall i \in \mathbb{A}_k^{u_{l,1}}\right) \wedge \left(\forall i \in \mathbb{A}_k^{u_{l,2}}\right) \wedge \cdots \wedge \left(\forall i \in \mathbb{A}_k^{u_{l,N^u}}\right)$$

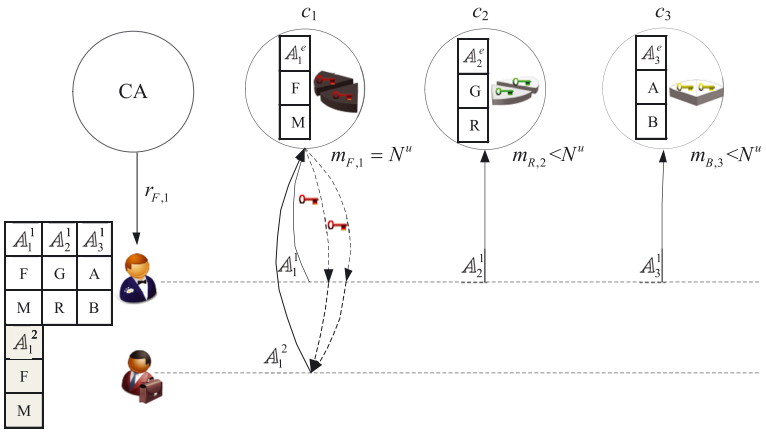
Finally, each one of legal users $\left\{u_{l,n}\right\}_{n\in[1,\ldots,N_c^u - 1]}$ has enough subkeys $\left\{S_{k,i}\right\}_{i\in\mathbb{A}_k^u} \supseteq \left\{S_{k,i}\right\}_{i\in\mathbb{A}_k^e}$, which are interpolated to recover m as follows:

$$\begin{aligned}
&E \Big/ \prod_{k=1}^{N^c}\prod_{n=1}^{d_k} \hat{e}\left(S_{k,i_n}, E_{k,i_n}\right)^{\lambda_k(n)} \\
&= E \Big/ \prod_{k=1}^{N^c}\prod_{n=1}^{d_k} \hat{e}(g_1, g_2)^{sp_k(i_n)\lambda_k(n)} \\
&= E \Big/ \prod_{k=1}^{N^c} \hat{e}(g_1, g_2)^{s\sum_{n=1}^{d_k} p_k(i_n)\lambda_k(n)} \\
&= E \Big/ \prod_{k=1}^{N^c} \hat{e}(g_1, g_2)^{sp_k(0)} \\
&= E \Big/ \hat{e}(g_1, g_2)^{s\sum_{k=1}^{N^c} p_k(0)} = E \Big/ \hat{e}(g_1, g_2)^{s\sum_{k=1}^{N^c} v_k} \\
&= E \Big/ Y^s = m
\end{aligned}$$

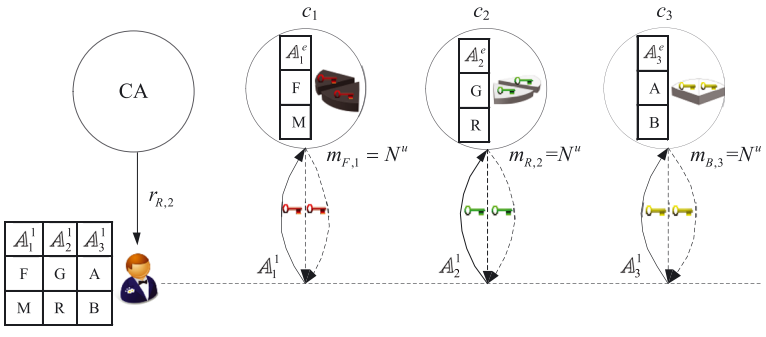
However, all colluders only have partial subkeys $\left\{S_{c,i}\right\}_{i\in\mathbb{A}_c^{u_a}} \subset \left\{S_{k,i}\right\}_{i\in\mathbb{A}_k^e}$, which are contributed by only u_a from AA c. Obviously, these subkeys are not enough to recover m.



(a) The first round.



(b) The second round.



(c) The last round.

Figure 2. Security check for the mixture of legal user and colluders.

Security Comm. Networks 2013; **6**:1143–1152 © 2013 John Wiley & Sons, Ltd.
DOI: 10.1002/sec

6. OUR SECOND SYSTEM

Our first system is vulnerable to denial-of-service attack launched by malicious user. And what is worse, the user may fool AA into issuing him or her subkeys by sending duplication request as if multiple users make requests. In the first system, it is assumed that any user including colluder submits his or her request only once. However, if the assumption is removed and no GID is used, it is hard to distinguish the duplication requests of one user from the same ones of different users. In this section, we construct an improved system to overcome this obstacle.

6.1. Attribute transformation

Because one request only corresponds to one AA instead of all AAs (based on the assumption that there are no overlaps between the attribute sets controlled by AAs), we consider that the same requests of different users are transformed into different ones by means of the subset of attributes indexes (defined in the set of encryptor's attributes). More specially, for $\{m_{i,k} * i\}_{i\in\mathbb{A}_k^e}$ formed by the request $\mathbb{A}_k^j$ from each user $j \in [1,\ldots,N_k^u]$ to each AA $k\in[1,\ldots,N^c]$, we map it into a normal set $\mathbb{L}_k^j$ including different element l_j, which satisfies congruence Equation (1) as follows:

$$l_j \bmod |\mathbb{A}_k^e| = L_i \tag{1}$$

where L_i is the index of i defined in $\mathbb{A}^e$.

6.2. Secret sharing based on discrete Fourier transform

In case decryption is dependent on the transformation set $\mathbb{L}_k^j$ of each user $j \in [1,\ldots,N_k^u]$ corresponding to each AA $k\in[1,\ldots,N^c]$, not only it is difficult for AA to exclude the invalid attribute (transformed by $i\notin\mathbb{A}_k^e$) from $\mathbb{L}_k^j$, but also each user j with the "same" $\mathbb{A}_k^j$ has "different" keys associated with $\mathbb{L}_k^j$, which is somewhat contradictory to ABE. To securely and efficiently recover a system-wide master secret key, we transform $\mathbb{L}_k^j$ using Equation (2) as follows:

$$f(x) = \exp(2\pi x\mathrm{i}/|\mathbb{A}_k^e|) \tag{2}$$

where i and exp(·) denote imaginary unit and exponential function with the base of the natural logarithm, respectively. More specially, each different $l_j \in \mathbb{L}_k^j$ is taken as the input of Equation (2) to generate $m_{i,k}$ numbers of the same unity root $\omega = \exp(2\pi L_i\mathrm{i}/|\mathbb{A}_k^e|)$. In this way, $\mathbb{L}_k^j$ is transformed into $\{m_{\omega,k} * \omega\}_{\omega\in\mathbb{W}_k^e}$, where $m_{\omega,k} = m_{i,k}$ and $\mathbb{W}_k^e = \{\omega_n\}_{n\in[1,\ldots,|\mathbb{A}_k^e|]}$.

Furthermore, a polynomial $p_k(x)$ of $d_k - 1 (d_k \le |\mathbb{A}_k^e|)$ degrees is constructed by Equation (3) to extract shares. Similar to Lagrange interpolation polynomial, each term coefficient $a_n \in \mathbb{Z}_\mu (0\le n\le d_k - 1)$ of $p_k(x)$ is randomly picked, and constant term coefficient a_0 is kept as a secret. The only change is that complex number (in the form of unity root) ω_n is taken as the input of $p_k(x)$. Obviously, the generation of shares $p_k(\omega_n)$ corresponds to perform DFT for d_k numbers of different unity roots ω_n.

$$p_k(x) = a_0 + \sum_{n=1}^{d_k-1} a_n x^n (\mathrm{mod}\mu) \tag{3}$$

Finally, a_0 is recovered by Equation (4), which corresponds to perform IDFT for d_k numbers of shares $p_k(\omega_n)$.

$$a_0 = \frac{1}{d_k}\sum_{n=0}^{d_k-1} p_k(\omega_n) (\mathrm{mod}\mu) \tag{4}$$

It is noted that ω_n is not directly used in the aforementioned recovery, which is different from Lagrange interpolation method where original attribute i is essential to interpolate shares.

6.3. Construction

For each user $j \in [1,\ldots,N_k^u]$ corresponding to AA $k\in[1,\ldots,N^c]$:

Setup. j transforms $\mathbb{A}_k^j$ into $\mathbb{L}_k^j$ using Equation (1), and system publishes the same SP as the respective one of the first system construction.

Encryption. Encryptor e publishes the same ciphertext C as the respective one of the first system.

Issuing Key. j submits $\mathbb{L}_k^j$ to k. Upon receipt of $\mathbb{L}_k^j$, k works as follows:

(1) Remove multiple duplications of $\mathbb{L}_k^j$ by comparing $\mathbb{L}_k^j$ with other requests.

(2) Transform $\mathbb{L}_k^j$ into $\{m_{\omega,k} * \omega\}_{\omega\in\mathbb{W}_k^j}$ using Equation (2).

(3) Call Algorithm 2 with the inputs of N^u and $\{m_{\omega,k} * \omega\}_{\omega\in\mathbb{W}_k^j}$.

(a) If Algorithm 2 outputs True, reject the n users' requests for subkeys.

(b) If Algorithm 2 outputs False, use Equation (3) to construct a $d_k - 1$ degrees polynomial $p_k(x)$ where $p_k(0) = v_k$. Let $\{t_{k,\omega}\}_{\omega\in\mathbb{W}_k^e} = \{t_{k,i}\}_{i\in\mathbb{A}_k^e}$, extract the set $\{S_{k,\omega} = g_1^{p_k(\omega)/t_{k,\omega}}\}_{\omega\in\mathbb{W}_k^j}$ of subkeys, and separately issue $S_{k,\omega}$ to the n users.

Decryption. The receiver j with $|\mathbb{W}_k^j| \ge d_k$ executes the following:

(1) Pair up S_{k,ω_n} and E_{k,i_n} to compute $\hat{e}(S_{k,\omega_n}, E_{k,i_n})$.

(2) Interpolate all values $\hat{e}(S_{k,\omega_n}, E_{k,i_n})$ by means of IDFT to compute Y^s as follows:

$$\prod_{k=1}^{N^c}\prod_{n=1}^{d_k}\hat{e}\left(S_{k,\omega_n},E_{k,i_n}\right)^{\frac{1}{d_k}}$$
$$=\prod_{k=1}^{N^c}\prod_{n=1}^{d_k}\hat{e}\left(S_{k,\omega_n},E_{k,\omega_n}\right)^{\frac{1}{d_k}}$$
$$=\prod_{k=1}^{N^c}\prod_{n=1}^{d_k}\hat{e}(g_1,g_2)^{\frac{sp_k(\omega_n)}{d_k}}$$
$$=\prod_{k=1}^{N^c}\hat{e}(g_1,g_2)^{\frac{s}{d_k}\sum_{n=1}^{d_k}p_k(\omega_n)(\mathrm{mod}\mu)}$$
$$=\prod_{k=1}^{N^c}\hat{e}(g_1,g_2)^{sp_k(0)}=\hat{e}(g_1,g_2)^{s\sum_{k=1}^{N^c}p_k(0)}$$
$$=\hat{e}(g_1,g_2)^{s\sum_{k=1}^{N^c}v_k}=Y^s$$

(3) Recover $m=E/Y^s$.

6.4. Security

Upon receipt of all requests in the form of $\{\mathbb{L}_k^j\}_{j\in[1,\ldots,N_k^u]}$, the use of $\mathbb{L}_k^j$ is impossible for each AA k to build a "complete profile" of one user because $\mathbb{L}_k^j$ is only related to a "local" set of attribute indexes. But in the case of collusion between all AAs, it is very likely to identity the user by "combined request", which can be randomly generated by linking all received requests across AAs. As more users participate in our system, it is more difficult for all AAs to identify the user because more combinations of requests are presented to AAs. Let Pr be the probability of building up a complete profile of one user by linking requests. According to different request modes, Pr has three cases as follows:

$$Pr(n,N^c)=\begin{cases}\left(\frac{1}{n}\right)^{N^c}, & m_{i,k}=N^u\\ \frac{1}{(n+1)n^{N^c}}, & 1<m_{i,k}<N^u\\ 0, & m_{i,k}=1\end{cases}$$

where n is the number of the users extracted in security check. In the case of legal requests, given $N^u=20$, we can input $n=20$ and $N^c=5$, such that $P_0=3.125\times10^{-5}\%$ (as shown in Figure 3(a)). That is, five numbers of AAs have an extreme low probability of building up a "complete profile" of one of twenty numbers of legal users. In the case of hybrid requests, it is shown in Figure 3(b) that we have similar results by choosing proper parameters. In the case of collusion requests, the request received by each AA is directly used to identify one user. But under the assumption that all requests are linked, it is impossible for AAs to build a complete profile of the user.

7. EFFICIENCY

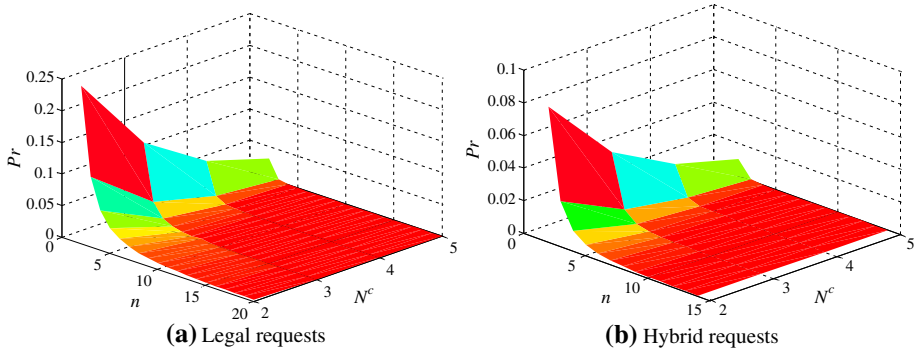
7.1. Size of key and ciphertext

In the schemes of Chase ABE and Chase–Chow ABE, AA stores $|\mathbb{A}_k^u|$ random values to construct AA key. In addition, one and N^c-1 random seeds are stored to define a hash function and PRF, respectively. Accordingly, one and N^c-1 special values are additionally stored by the user to unlink sub keys with GID and GID-related PRF term, respectively. However, in our scheme, each AA k only stores $|\mathbb{A}_k^u|$ values (note that N^u is only used for security checking instead of generating key), and the user in turn only stores $|\mathbb{A}_k^u|$ values. Therefore, it is shown in Figure 4(a) that the sizes of our AA key and user key are smaller than those of the schemes of Chase ABE and Chase–Chow ABE.

Moreover, it is shown in Figure 4(b) that our scheme publishes a smaller ciphertext, compared with Chase–Chow ABE scheme because our ciphertext is in the form of a two-tuple rather than three-tuple in Chase–Chow ABE scheme.

7.2. Costs of key issue and decryption

The main overhead is on the side of the authority. For three request modes mentioned earlier, we compare the cost of our scheme with the respective one of existing schemes. In the case of collusion requests, our scheme incurs negligible overhead resulting from one round of a security check, which prevents each AA from issuing any subkeys



(a) Legal requests **(b)** Hybrid requests

Figure 3. Probability of privacy leakage resulting from linking requests.

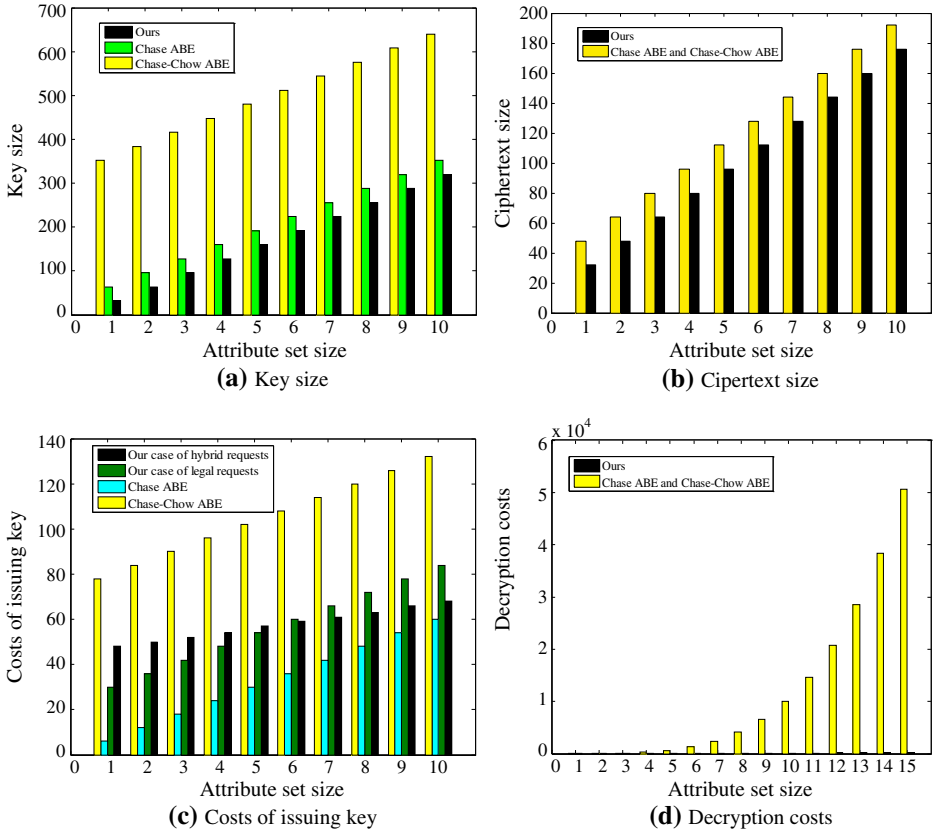


Figure 4. Efficiency results for our scheme and existing attribute-based encryption schemes.

to colluders. In the cases of legal and hybrid requests, it is shown in Figure 4(c) that the cost of our scheme is more than that of Chase ABE scheme but is obviously less than that of Chase–Chow ABE scheme. Because our scheme excludes most colluders from request users by means of security check, which runs only one interaction (between each user and each AA) for legal request, and at most three interactions for hybrid request, while four interactions are required in anonymous key issuing protocol of Chase–Chow ABE scheme.

In decryption, the size of $\mathbb{A}_c^u$ has a direct impact on computation cost—increasing $\left|\mathbb{A}_c^u\right|$ incurs more operations of addition and multiplication, and decreasing $\left|\mathbb{A}_c^u\right|$ has the opposite effect. It is shown in Figure 4(d) that our scheme has lower computation cost, compared to the schemes of Chase ABE and Chase–Chow ABE. Because the time complexity $O\left(\left|\mathbb{A}_c^u\right|^2\right)$ of IDFT (adopted by our scheme) is less than $O\left(\left|\mathbb{A}_c^u\right|^4\right)$ of Lagrange interpolation (adopted by the schemes of Chase ABE and Chase–Chow ABE).

8. FUTURE DIRECTIONS

There are still a number of practical problems to be studied. We leave them as a future work for defending against multi-collusion attack, which is simultaneously launched by a number of cliques. In our system, we consider that only one clique launches attack on ABE. Thus, the malicious user out of the clique needs to communicate with the clique to copy subkeys. However, in large-scale distributed networks, it is more convenient for multiple malicious users to form another clique for themselves, compared to link with latent clique from long distance. Obviously, this multi-collusion will heavily disturb AA in security check for user. The second practical problem is that a trusted CA has power to prevent user from participating in the request for key, which seems somehow contradictory to the original goal of distributing control over many potentially untrusted authorities. Therefore, how to design a fully distributed ABE scheme against multi-collusion attack needs further investigation.

ACKNOWLEDGEMENTS

This work was partially sponsored by National Natural Science Foundation of China (NSFC Key Project on Privacy Preserving Model and Key Issues in Pervasive Environment) with grant number 60873262.

REFERENCES

1. Shamir A. Identity-based cryptosystems and signature schemes. Proc. International Cryptology Conference, Aug. 1984; 47–53.
2. Sahai A, Waters B. Fuzzy-identity based encryption. Proc. 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, May 2005; 457–473.
3. Cheung L, Newport C. Provably secure ciphertext policy ABE. Proc. 14th ACM Conference on Computer and Communications Security, Oct. 2007; 456–465.
4. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. Proc. IEEE Symposium on Security and Privacy, May 2007; 321–334.
5. Kapadia A, Tsang PP, Smith SW. Attribute-based publishing with hidden credentials and hidden policies. Proc. 14th Network and Distributed System Security Symposiom (NDSS), Feb. 2007.
6. Nishide T, Yoneyama K, Ohta K. Attribute-based encryption with partially hidden encryptor-specified access structures. Proc. 6th Applied Cryptography and Network Security (ACNS), June 2008; 111–129.
7. Chase M. Multi-authority attribute based encryption. Proc. 4th Theory of Cryptography Conference, Feb. 2007; 515–534.
8. Chase M, Chow SSM. Improving privacy and security in multi-authority attribute-based encryption. Proc. 16th ACM Conference on Computer and Communications Security, Nov. 2009; 121–130.
9. Dodis Y, Yampolskiy A. A verifiable random function with short proofs and keys. Proc. 8th International Workshop on Theory and Practice in Public Key Cryptography, Jan. 2005; 416–431.
10. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. Proc. 13th ACM Conference on Computer and Communications Security, Oct. 2006; 89–98.
11. Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. Proc. 14th ACM Conference on Computer and Communications Security, Oct. 2007; 195–203.
12. Smith SW. *The Scientist and Engineer's Guide to Digital Signal Processing*, (2nd edn.). California Technical Publishing: San Diego, California, 1999.