Computing and Informatics, Vol. 35, 2016, 128–142

SIMPLE MULTI-AUTHORITY ATTRIBUTE-BASED ENCRYPTION FOR SHORT MESSAGES

Viktória I. Villányi

Department of Operations Research
ELTECRYPT Research Group
Eötvös Loránd University
Pázmány Péter Sétány 1/C
1117 Budapest, Hungary
e-mail: vvillanyi@gmail.com

Abstract. Central authority free multi-authority attribute based encryption scheme for short messages will be presented. Several multi-authority attribute based encryption schemes were recently proposed. We can divide these schemes into two groups, one of them are the ciphertext-policy attribute based encryption schemes (CP-ABE), the another one are the key-policy attribute based encryption schemes (KP-ABE). In our new multi-authority attribute based encryption scheme we combine them: the access structure will be given by authorities and the encryptor in conjunction. The authorities will be able to decide who is able to decrypt a ciphertext under their names, but the encryptor will choose the authorities whom he would involve in the encryption. In addition, our scheme is free of any central authority. The security of our new scheme relies on the decisional 3-party Diffie-Hellman assumption.

Keywords: Multi-authority attribute based encryption, key policy, ciphertext policy, central authority free

Mathematics Subject Classification 2010: 94A60

1 INTRODUCTION

The basic idea of attribute based schemes is based on Shamir's identity-based cryptosystems [14]. Boneh and Franklin [1] built the first identity-based encryption scheme that used pairings and its security depends on the bilinear Diffie-Hellman assumption. Sahai and Waters in [15] introduced the concept of attribute-based encryption. In an attribute-based encryption system the users' private keys and

ciphertexts are associated with a set of attributes or a policy over the attributes. The user is able to decrypt if the pairs between his secret keys and the ciphertext satisfy a given access structure. Several attribute-based encryption schemes were introduced in the recent years. We can categorize these schemes into two groups, one of the groups is the Key-Policy Attribute-Based Encryption (KP-ABE) (introduced in [8]), the other one is the Ciphertext-Policy Attribute Based Encryption (CP-ABE) (introduced in [4]). In the KP-ABE case, the access structure is defined by the authorities and it is integrated into the secret key; the user will be able to decrypt if and only if the ciphertext associated with his attributes satisfies the key policy. In the CP-ABE case, the access structure is defined by the encryptor and it is integrated into the ciphertext; the user will be able to decrypt if and only if his attributes corresponding secret keys satisfy the ciphertext's policy.

Chase designed the first multi-authority attribute-based encryption scheme [7]. In her paper she introduced a trusted central authority who has to stay fully trusted along the lifetime of the system. The central authority combines the authorities' public keys and it produces the public key of the system. Later Božović et al. in [2], Chase and Chow in [6] and Lin et al. in [9] improved her system: they built multi-authority encryption schemes without a fully trusted central authority. These schemes belong to the KP-ABE schemes since the access structure is given in secret keys.

Müller et al. in [12] proposed the distributed attribute-based encryption schemes as an extension of CP-ABE schemes. Their scheme supports an arbitrary number of attribute authorities and allows to dynamically add new users and authorities to the system at any time. Their scheme still assumes a central authority, who has the master secret key of the system and is able to decrypt any message.

In 2011, Lewko and Waters [10] designed a CP-ABE multi-authority encryption scheme without any central authority. In their scheme the authorities do not even need to be aware of each other but their scheme has its limitation; the authorities are able to output one attribute per public key. In the appendix they suggested a repeated use of the same protocol to solve this problem, this solution increased the size of keys and attributes, that is a significant loss in efficiency. Lewko and Waters improved their encryption scheme [11]; the new scheme allows unrestricted use of attributes and proven to be secure in the standard model.

We should mention the functional encryption schemes [5], as a generalization of the attribute based encryption schemes. That is a very recent research topic in the public key cryptography. In functional encryption schemes, ciphertexts are associated with values x and the secret keys are associated with values y, and the function $f(x, y)$ defined what the user with secret key y should learn about the ciphertext x.

Our contribution. Here we present a combo encryption scheme that is free of central authority and combines the advantages of KP-ABE and CP-ABE schemes. No communication between authorities is needed, and the authorities do not even have to be aware of each other like in the above mentioned Lewko-Waters

scheme. In our scheme the authorities still will be capable to influence who is able to decrypt a ciphertext that is encrypted under their names (KP-ABE), but also an encryptor will be able to choose the set of authorities whose attributes will be needed to decrypt a ciphertext (CP-ABE). The scheme is very dynamic, the encryptor can chose any subset of authorities he wants to involve in the encryption. The most important feature of our scheme is that there is no central entity in our system who is capable to decrypt (possess the master secret key) any message.

Collusion challenge. The greatest challenge in the multi-authority attribute based encryption schemes is to avoid collusion between users. By the collusion resistance we usually mean that the users are unable to decrypt a ciphertext by combining their secret keys if they are not capable to decrypt a ciphertext solely based on their own secret keys. We use a global identifier to distinguish users and to avoid collusion between users. Global identifiers GID_us in multi-authority encryption schemes were suggested by Chase in [7]. We define a hash function on the user's GID_u, the hash function behaves like a random oracle and outputs $H(GID_u) \in_R G_1$ element for user u. The authorities use these values to output the users' secret keys by generating a secret sharing polynomial for every user with a constant term that equals $H(GID_u)$. Since $H(GID_u)$ are chosen randomly the collusion resistance between users is achieved. We also need to be certain that a user cannot combine his own secret keys that he received from different authorities and decrypt a ciphertext if he does not have enough secret keys from all involved authorities. It could cause a problem since his secret keys belong to polynomials which has $H(GID_u)$ at 0 even if they were built by different authorities. These $H(GID_u)$ values will also serve as a basis for the user to be able to combine his attributes from different authorities and decrypt the ciphertext. The collusion resistance problem in this case is similar to the problem we had in the previous attribute-based encryption schemes for one authority when the polynomial value at 0 was the same value for every user and we wanted to prevent collusion between users.

2 PRELIMINARIES

Some basic knowledge on bilinear maps, decision 3-party Diffie-Hellman assumption and multi-authority attribute based encryption schemes is necessary to build our scheme. We give a short introduction into these topics in the following section. We also define our message space and a hash function on the set of global identifiers $\{GID_u\}$.

2.1 Bilinear Maps and the Decision 3-Party Diffie-Hellman Assumption

Let G_1 and G_2 be groups of prime order p and P be a generator of G_1. We use the additive notation for G_1 and the multiplicative notation for G_2. We denote by

$e : G_1 \times G_1 \to G_2$ an *admissible bilinear map* if all of the following requirements hold:

1. for all $P, Q \in G_1$ and for all $a, b \in \mathbb{Z}_l^*$ $e(aP, bQ) = e(P, Q)^{ab}$,
2. $e(P, P) \neq 1_{G_2}$, i.e. $\langle e(P, P) \rangle = G_2$,
3. $e(P, Q)$ is computable in probabilistic polynomial time for arbitrary $P, Q \in G_1$.

The security of our scheme relies on the *decision 3-party Diffie-Hellman assumption.*

2.2 Decision 3-Party Diffie-Hellman(D3DH) Assumption

Given a group G_1 of prime order p with a generator P and random elements, $A = aP$, $B = bP$, $C = cP \in_R G_1$. We say the D3DH problem is hard relative to G_1 if for all probabilistic polynomial-time algorithm $\mathcal{A}$ there exists a negligible function $negl(n)$ such that

$$|Pr[\mathcal{A}(G, P, aP, bP, cP, abcP) = 1] - Pr[\mathcal{A}(G, P, aP, bP, cP, rP) = 1]| \leq negl(n).$$

Remark 1. D3DH assumption implies the decision billinear Diffie-Hellman assumption.

If an adversary is able to break the system, then we are able to construct a *Simulator* that breaks the D3DH assumption.

2.3 Message Space

We use global identifier to distinguish users and to achieve the collusion resistance. Every user is acquainted with a user specific GID_u value. The *System* uses a hash function $H(GID_u) \to G_1$, that maps the users's GID_u into a random element of group G_1. The hash function H is given in the setup phase.

To perform encryption and decryption it is necessary that the bilinear map e to be preimage resistant for $e(R, H(GID_u)) \in G_2$ where $R \in_R G_1$ chosen uniformly at random from G_1 but not to be preimage resistant for $e(M, H(GID_u)) \in G_2$, where the message $M \in_R \mathbb{M} \subset G_1$ is chosen from the message space. By the preimage resistance we mean if $H(GID_u) \in G_1$ and $W \in G_2$ are given then it is hard to find R such that $e(R, H(GID_u)) = W$ holds.

To solve this problem we use lookup tables. The message space is a list of all possible messages and it is given in the *Setup* phase. Every user has to build his own lookup table that contains all possible messages from the message space $M \in \mathbb{M} \subset G_1$, and the $e(M, H(GID_u)) \in G_2$ values where $H(GID_u) \in_R G_1$ is the hash of the user's identifier.

The table contains the list of these pairs:

$$M \in G_1 \to e(M, H(GID_u)) \in G_2.$$

2.4 Authorities, Attributes and Users

Let $\mathcal{K}$ be the set of authorities, $\mathcal{U}$ be the set of users, $\mathcal{A}$ be the set of attributes. Let $\mathcal{A}_k \subset \mathcal{A}$ be the set of attributes handled by authority $k \in \mathcal{K}$. The authority k issues the secret key to user u for attribute a (handled by authority k) if the user u shows his valid credential for this attribute to him. We denote the set of attributes by $\mathcal{A}_u$ if user u received his attribute specific secret key. The adversary has to specify a set of authorities and a set of their attributes on which he wants to be challenged. We denote this challenge set of attributes by $\mathcal{A}_C$. The authority $k \in \mathcal{K}$ chooses a d_k threshold; d_k is the number of the attributes secret key pairs (issued by $\mathcal{A}_k$) that are needed to decrypt a ciphertext. In the multi-authority settings a user is able to decrypt the message if he has at least d_k attributes from every authority that were involved in the encryption

$$|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_C| \geq d_k \text{ for every } k \in \mathcal{A}_k.$$

The *multi-authority attribute-based encryption scheme* is a tuple of the following probabilistic polynomial time algorithms:

Setup. It is a randomized algorithm run by a trusted party that takes the security parameter 1^l as input and returns the public system parameters.

Attribute secret key generation. A randomized algorithm that takes the public system parameters which includes the users' global identifier GID_u, a hash function defined on the GID_us, the authorities, threshold d_k, and the set of attributes as input, and it returns the secret keys for user u.

Encryption. A randomized algorithm run by the encryptor that takes a given message, access structure, authorities with their attributes $\mathcal{A}_C \subseteq \mathcal{A}$, the public system parameters as input, and it outputs a ciphertext.

Decryption. A deterministic algorithm takes the user's (u) secret keys for attribute set $\mathcal{A}_u$, a ciphertext, and the set of attribute that were used in the encryption ($\mathcal{A}_C$) as input, and it returns the plaintext (M) if $M \in \mathbb{M}$ and $|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_C| \geq d_k$ for every authority that was chosen by the encryptor; otherwise it outputs an error symbol $\perp$.

3 SECURITY MODEL

We prove security in the Selective ID model. This model is used in several papers on multi-authority attribute based encryption schemes [7, 3, 9]. In this model the adversary has to specify an identity (a set of attributes), that he wishes to be challenged upon before the public keys of the system are generated.

The ***Security game*** of our multi-authority attribute based encryption scheme in the *selective ID model*.

Setup

1. Given a security parameter 1^l, the adversary outputs
   - set of users $\mathcal{U}$
   - set of attribute authorities $\mathcal{K}$
   - the challenge identity $\mathcal{A}_\mathcal{C} \subset \mathcal{A}$
2. The public and secret keys are generated, the adversary will learn:
   - The public system parameters which includes attributes (values) of the authorities, the users' identifiers ($GID_u s$), the hash function, the list of possible messages, G_1, G_2 groups and a bilinear map.

Secret key queries
The adversary $\mathcal{M}$ issues queries for secret keys of users but he has the following restrictions on the secret key queries:

- for each user u there is at least one authority $\hat{k}_u$ for which the adversary $\mathcal{M}$ has less than $d_{\hat{k}_u}$ secret keys.
- for each user u no authority $k \in \mathcal{K}$ is queried more than once for secret keys of user u.

Challenge Phase

1. The adversary $\mathcal{M}$ outputs two messages M_0, M_1 from the message space $\mathbb{M}$.
2. The challenger chooses $\mu \leftarrow \{0,1\}$ uniformly at random and outputs the encryption of M_μ using the challenge attribute set A_C.
3. The ciphertext C is given to the adversary $\mathcal{M}$.

Secret key queries
The adversary can issue further secret key queries with the same restrictions as before.

Guess
The adversary guesses which message (M_μ) has been encrypted. His guess for μ is $\mu' \in \{0,1\}$.

The advantage of the adversary in this game:

$$Adv_M^{sid}(l) := Pr(\mu' = \mu) - \frac{1}{2}.$$

Definition 1. A multi-authority attribute-based encryption is secure, if for all probabilistic polynomial time adversaries $\mathcal{M}$ the advantage $Adv_M^{sid}(l)$ is negligible.

4 OUR NEW PROTOCOL

We begin this section with a brief overview of our protocol, then we give a more detailed description of our multi-authority attribute-based encryption scheme.

4.1 Overview of the Protocol

At first the *system parameters* need to be generated. These parameters include G_1 and G_2 prime order groups with an admissible bilinear map (e), a hash function defined on GID_us, a generator $\langle P \rangle$ of G_1, and the list of all possible messages $\mathbb{M}$. From the list of messages the users build their own lookup tables. After the *system parameters* generation the authorities generate the *attributes' public keys* and the *users's secret keys*. The attributes' public keys are public and represent the real attributes in the system. We use a threshold secret sharing scheme which is based on Shamir's (n, k)-threshold secret sharing scheme[13] to generate secret keys. The authorities perform the secret key generation by building a random secret sharing polynomial for every user u with a constraint that the constant term of these polynomials is $H(GID_u)$. The authority $\mathcal{A}_k$ handouts the shares, the user's secret keys, to qualified users, and it publishes the set of attributes. If the user received enough secret keys, $|A_C \cap A_k \cap A_u| \geq d_k$, then he is able to recover his $e(P, H(GID_u))$ value by Langrange interpolation. The product of the $e(P, H(GID_u))$ values on some random power (used in the encryption) hides the message. The $e(P, H(GID_u))$ values behave like the public keys.

In the encryption process the encryptor chooses a set of authorities whom he wants to involve in the encryption. The encryptor chooses $r_i \in_R \mathbb{Z}_p$ uniformly at random for every involved authority. The encryptor multiplies the involved attributes of authority i by value r_i. Attributes from the same authority are multiplied by the same r_i value, attributes from different authorities are multiplied by different r_i values. He also computes $rP = \sum_i r_i P$ value to hide the plaintext. The encrypted message is $rP + M$. If the user possesses at least d_k secret keys of all the chosen authorities, then he is able to decrypt the message by computing

$$e(P, H(GID_u))^r = \prod_i e(P, H(GID_u))^{r_i}$$

and by removing the hiding factor of the message with the following computation

$$\frac{e(Enc(M), H(GID_u))}{\prod e(P, H(GID_u))^{r_i}} = \frac{e(rP + M, H(GID_u))}{e(P, H(GID_u))^r} = e(M, H(GID_u)).$$

From $e(M, H(GID_u))$ the user is able to recover the message M by using his look up table. If $e(M, H(GID_u))$ is not contained in his lookup table, then the ciphertext is invalid.

4.2 The Proposed Protocol Step by Step

We give a more detailed description of our new protocol here. We begin it with the system parameters generation.

System parameters

- G_1 and G_2 prime order groups
- $\langle P \rangle = G_1$, the generator of group G_1
- e the admissible bilinear map
- H hash function defined on GID_us
- $\mathbb{M}$ the list of all possible messages

Attributes generation
Attribute authority $k \in \mathcal{K}$ chooses for each of its attributes $a \in \mathcal{A}_k$ a secret value $t_{k,a} \leftarrow (\mathbb{Z}/p\mathbb{Z})^*$ uniformly at random and outputs

$$\underbrace{[t_{k,a} \cdot P]}_{=:T_{k,a}}{}_{a \in \mathcal{A}_k}$$

the attribute representing value. We call these values for attributes in our *system*. The authorities' secret key contains the $[t_{k,a}]_{a \in \mathcal{A}_k}$ values.

Attribute key generation
For each user $u \in \mathcal{U}$, the attribute authority k chooses uniformly at random a secret polynomial $f_{k,u} \in \mathbb{F}_p[X]$ of degree $< d_k$ with a constraint $f_{k,u}(0)P = H(GID_u)$.

$$f_{k,u}(x)P = H(GID_u) + a_{1_{k,u}} xP + a_{2_{k,u}} x^2 P + \cdots + a_{d_{k_{k,u}}-1} x^{d_k - 1} P$$

In order to generate secret keys for users, we assume that each attribute $a \in \mathcal{A}$ can be identified with a unique number $\iota(a) \in \{1, \ldots, p-1\}$. To create the attribute secret keys for user $u \in \mathcal{U}$ associated with an attribute $a \in \mathcal{A}_k \cap \mathcal{A}_u$, the attribute authority $\mathcal{A}_k$ computes the attribute and user specific value $D_{k,u,a} := \frac{f_{k,u}(\iota(a))}{t_{k,a}} \cdot P$. If the user has the necessary credentials for attribute a, then the authority gives the corresponding $D_{k,u,a}$ secret key to the user.

Encryption
To encrypt a message $M \in \mathbb{M}$, the encryptor picks a set of authorities $A_{\mathcal{S}}$, $\mathcal{S} \subseteq \mathcal{K}$, whose attribute sets' subset will be used in the encryption; $d := |\mathcal{S}|$ is the number of them. We denote the set of attributes that are used in the encryption by $\mathcal{A}_C$; it is a subset of the chosen authorities' attributes. The encryptor also chooses $\rho_i \leftarrow \{0, \ldots, p-1\}$ uniformly at random and computes the ciphertext as follows,

$$CT_{\mathcal{A}_\mathcal{C}} = (\mathcal{A}_\mathcal{C}, [\rho_k \cdot T_{k,a}]_{a \in \mathcal{A}_C}, \rho P + M)$$

with

$$\rho = \sum_{i=1}^{d} \rho_i.$$

Decryption
Let $CT_{\mathcal{A}_\mathcal{C}} = (\mathcal{A}_\mathcal{C}, [\rho_k \cdot T_{k,a}]_{a \in \mathcal{A}_C}, \rho P + mP)$ be a ciphertext with the associated attribute set $\mathcal{A}_C$. If user u's attribute set $\mathcal{A}_u$ satisfies $|\mathcal{A}_u \cap \mathcal{A}_k| \geq d_k$ for all $k \in \mathcal{S}$, then user is able to recover the plaintext M as follows.

1. For each $k \in \mathcal{S}$, user u chooses d_k attributes $a \in \mathcal{A}_u \cap \mathcal{A}_k$ and computes

$$e(\rho_k \cdot T_{k,a}, D_{k,u,a}) = e(P, P)^{f_{k,u}(x) \cdot \rho_k}$$

Then by using Lagrange polynomial interpolation, the user u computes

$$e(P, H(GID_u))^{\rho_k}.$$

2. And he computes the hiding factor of the message.

$$e(P, H(GID_u))^{\rho} = \prod_{k=1}^{d} e(P, H(GID_u))^{\rho_k}$$

To decrypt a message the user u takes the encrypted message, $Enc(M) = \rho P + M$, and computes $e(\rho P + M, H(GID_u))$. Then he divides it with the previously computed $e(P, H(GID_u))^{\rho}$ value.

$$\frac{e(\rho P + M, H(GID_u))}{e(P, H(GID_u))^{\rho}} = e(M, H(GID_u))$$

From $e(M, H(GID_u))$ by using the lookup table the user is able to recover the plaintext M. If the users' look up table does not contain $e(M, H(GID_u))$, then the ciphertext is invalid.

5 SECURITY PROOF

We prove the security of our scheme by contradiction in the Seletive-ID model. We suppose our scheme is not secure, then there exists an adversary $\mathcal{M}$ that is able to break the scheme. If $\mathcal{M}$ exists, then we are able to build a *Simulator* S that succeed in breaking the D3DH assumption, that leads to a contradiction.

Theorem 1. If there exists a probabilistic polynomial time adversary $\mathcal{M}$ that has non-negligible advantage in our security game, then there is a probabilistic time algorithm *Simulator* that has a non-negligible advantage in solving the Decision 3-Party Diffie-Hellman problem.

Proof. The input of *S(imulator)* algorithm is a tuple

$$(P, A, B, C, (\delta \cdot abcP + (1-\delta)rP)),$$

where $A = aP,\ B = bP,\ C = cP$ are chosen uniformly at random from G_1, and $\delta \leftarrow \{0,1\}$ is chosen uniformly at random. S simulates the attribute authorities to *Adversary* $\mathcal{M}$ and answers the secret key queries. The protocol created by the *Simulator* is indistinguishable from the regular protocol. *Simulator* uses the knowledge of *Adversary* $\mathcal{M}$ to find δ. □

Simulation of the public parameter generation
The *Simulator* generates the hash of the users' global identifier by choosing $l_u \in_R \mathbb{Z}_p$ uniformly at random and outputs $H(GID_u) := l_u P$ for user u. Since we are proving security in the Selective ID model, the adversary outputs a set of attributes, $\mathcal{A}_C$ on which set he wants to be challenged on, at first. The *Simulator* generates the authorities' attributes. He chooses $t_{k,a} \in_R \mathbb{Z}_p$ values uniformly at random, for the k^{th} authority's attribute a, and he assigns the following $T_{k,a}$ values to attributes:

- $[t_{k,a}P]$ if $a \in \mathcal{A}_C$
- $[t_{k,a}B]$ if $a \in \mathcal{A} \setminus \mathcal{A}_C$.

The *Simulator* publishes the $\{T_{k,a}\}_{a \in \mathcal{A}_\mathcal{C}}$ attributes and $B = bP$ as group generator. The *Adversary* can issue queries for secret keys of users. *Simulator* generates the secret keys of users by choosing random polynomials $(f_{k,u})$ for users u, and for authority k, with the constraint that $f_{k,u}(0) := l_u$. If d_k attributes are needed to decrypt under $\mathcal{A}_k$'s name, then S chooses $d_k - 1$ values uniformly at random $(s_{k,u,a} \in_R \mathbb{Z}_p)$ and he also add $f_{k,u}(0) = l_u$ value for user u to build $f_{k,u}$ polynomial of degree of $d_k - 1$. From these $s_{k,u}$ values and from $f_{k,u}(0) = l_u$ value the *Simulator* is able to compute the value of the polynomial $(f_{k,u}(x))$ at any point (x) by using interpolation, so the *Simulator* is capable to calculate the rest of the secret keys of user u.

Challenge
The *Adversary* outputs two messages $M_0, M_1 \in \mathbb{M}$. The *Simulator* flips a fair binary coin $\mu \leftarrow \{0,1\}$ and returns the challenge ciphertext.

The *Simulator* chooses uniformly at random $z_1,\ z_2, \ldots z_d$ values, with $d = |\mathcal{S}|$, where $\mathcal{S}$ is the number of authorities that are involved in the encryption. Then the *Simulator* compute $z := \sum_{k=1}^{d} z_k$ and outputs the ciphertext.

$$CT_{\mathcal{A}_\mathcal{C}} = (\mathcal{A}_\mathcal{C}, \{z_k t_{k,a} A\}_{a \in \mathcal{A}_\mathcal{C}}, (\delta \cdot abc + (1-\delta) \cdot r)P + M_\mu),$$

where $A = aP$. This is a valid encryption of M_μ for $\delta = 1$. The *Simulator* uses the knowledge of adversary $\mathcal{M}$ to find $\delta \leftarrow \{0,1\}$ value to break Decision 3-Party Diffie-Hellman assumption. At first we show that the *Simulator* is able to answer the *Adversary's* secret key queries.

Secret key queries
The attacker $\mathcal{M}$ can query the secret keys of any user with the following constraints: for every user there is at least one authority $\hat{k}$ from which the attacker cannot obtain more than $d_{\hat{k}} - 1$ secret keys. The adversary can query secret keys of user u from authority k at most once. The *Simulator* must be capable answering secret key queries.

We distinguish 2 cases:

- The secret keys correspond to an attribute from the challenge set $\mathcal{A}_{\mathcal{C}}$
  - If $|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_c| \geq d_k$ or earlier there has been a secret key query for user u such that $|\mathcal{A}_u \cap \mathcal{A}_{\hat{k}} \cap \mathcal{A}_c| \leq d_{\hat{k}}$ $k \neq \hat{k}$, then the *Simulator* outputs the $\frac{f_{k,u}(\iota(a))}{t_{k,a}} B$ values, where $B = bP$ is a part of the D3DH challenge and $\iota(a)$ is attribute's unique identifier.
  - If $|\mathcal{A}_u \cap \mathcal{A}_{\hat{k}} \cap \mathcal{A}_c| < d_{\hat{k}}$ and there has not been a previous secret key query for user u such that $|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_c| \leq d_k$ and $k \neq \hat{k}$ then the *Simulator* builds a new polynomial $f^*_{\hat{k},u}$ from the previously chosen $s_{\hat{k},u}$ values but he changes the value of the polynomial at 0 as follows:

$$f^*_{\hat{k},u}(0) := l_u \cdot \frac{C^*}{z_{\hat{k}}}, \qquad C^* := C - \sum_{i=1, i\neq\hat{k}}^{d} z_i P.$$

    Simulator outputs the $\frac{f^*_{\hat{k},u}(\iota(a))}{t_{\hat{k},a}} B$ value, where $f^*_{\hat{k},u}(\iota(a)) := s_{\hat{k},u,a}$ for the secret key of user u, for attribute a, from authority $A_{\hat{k}}$. The *Simulator* is able to hand out at most $d_{\hat{k}} - 1$ secret keys for user u's attributes from $\mathcal{A}_{\mathcal{C}} \cap \mathcal{A}_{\hat{k}}$.

- The secret keys correspond to an attribute from $\mathcal{A} \setminus \mathcal{A}_{\mathcal{C}}$
  - If $|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_c| \geq d_k$ or earlier there has been a secret key query for user u such that $|\mathcal{A}_u \cap \mathcal{A}_{\hat{k}} \cap \mathcal{A}_c| \leq d_{\hat{k}}$ $k \neq \hat{k}$ then the *Simulator* outputs the $\frac{f_{k,u}(\iota(a))}{t_{k,a}} P$.
  - If $|\mathcal{A}_u \cap \mathcal{A}_{\hat{k}} \cap \mathcal{A}_c| < d_{\hat{k}}$ and there has not been a previous secret key query for user u such that $|\mathcal{A}_u \cap \mathcal{A}_k \cap \mathcal{A}_c| \leq d_k$ and $k \neq \hat{k}$ then the *Simulator* outputs the $\frac{f^*_{k,u}(\iota(a))}{t_{k,a}} P$ values. The adversary is allowed to query less than $d_{\hat{k}}$ number of secret keys from authority $\mathcal{A}_{\hat{k}}$.

The adversary $\mathcal{M}$ can continue to *query users' secret keys* with the same restrictions as before.

Reconstruction of the hiding factor of the message for user u

$$e\left(z_{\hat{k}} t_{\hat{k},a} A, \frac{b l_u \left(C - \sum_{i=1, i\neq\hat{k}}^{d} z_i P\right)}{z_{\hat{k}} t_{\hat{k},a}}\right) \prod_{k=1, k\neq\hat{k}}^{d} e\left(z_k t_{k,a} A, \frac{bH(GID_u)}{t_{k,a}}\right)$$

$$= \frac{e(A, bl_uC)\prod_{k=1,k\neq\hat{k}}^{d} e(P, H(GID_u))^{abz_k}}{\prod_{i=1,i\neq\hat{k}}^{d} e(z_iP, H(GID_u))^{ab}} = e(P, l_uP)^{abc} = e(P, H(GID_u))^{abc}$$

Guess of the Simulator

The *Simulator* can take advantage of the knowledge of the adversary $\mathcal{M}$ and he can break the D3DH assumption as follows. After the second phase of the secret key queries the adversary $\mathcal{M}$ submits his guess μ' for μ. The *Simulator* will use μ' to output his guess δ' for δ.

$$\delta' := \begin{cases} 1, & \text{if } \mu = \mu' \\ 0, & \text{if } \mu \neq \mu'. \end{cases}$$

If the adversary $\mathcal{M}$ incorrectly guesses μ ($\mu \neq \mu'$) and rP were used in the encryption, then the probability that $\mathcal{S}$ returns a correct guess for δ is

$$Pr\left(\delta' = \delta | \delta = 0\right) = \frac{1}{2}.$$

If the adversary $\mathcal{M}$ correctly guesses μ ($\mu = \mu'$) and $abcP$ were used in the encryption, then the probability that $\mathcal{S}$ returns a correct guess for δ is

$$Pr(\delta' = \delta | \delta = 1) = Pr(\mu' = \mu | \delta = 1) = \frac{1}{2} + \mathrm{Adv}^{\mathrm{sid}}_{\mathcal{M}}(l)$$

If we combine these conditional probabilities, then we can calculate the advantage of the *Simulator* in solving a D3DH challenge:

$$\mathrm{Adv}^{D3DH}_{\mathcal{S}}(l) = Pr(\delta' = \delta) - \frac{1}{2}$$

$$\mathrm{Adv}^{D3DH}_{\mathcal{S}}(l) = \frac{1}{2} \cdot \left(\frac{1}{2} + \frac{1}{2} + Adv^{\mathrm{sid}}_{\mathcal{M}}\right)(l) - \frac{1}{2}$$

$$\mathrm{Adv}^{D3DH}_{\mathcal{S}}(l) = \frac{1}{2} \cdot Adv^{\mathrm{sid}}_{\mathcal{M}}(l)$$

6 COMMENTS ON OUR NEW PROTOCOL

Our new protocol offers several nice features. We show its modifiability and flexibility in this section.

6.1 Comment 1

We can slightly modify our system and derive another scheme with similar assumptions and security proof. Our secret key generating polynomial value at 0 was $H(GID_u)$. We could give more flexibility to the authorities by letting them to choose their own secret values a^*_k and change the value of the polynomial at 0 for $a^*_k H(GID_u)$. The authorities $\mathcal{A}_k$ have to use the same a^*_k for every user along the

lifetime of the system to build the user's secret keys. They also need to publish their $a_k^* P \in G_1$ values. The encryptor has to use these values to encipher the message but he needs to chose only one value ρ uniformly at random in $\mathbb{Z}_p$, and multiply every attributes by ρ. The ciphertext would be:

$$\left(CT_{\mathcal{A}_\mathcal{C}} = \mathcal{A}_\mathcal{C}, \rho T_{k,a}, \rho \left(\sum_{k \in \mathcal{A}_\mathcal{C}} a_k^* P \right) + M \right).$$

The decryption process would stay unchanged.

6.2 Comment 2

Another way to improve our basic scheme and derive a new more flexible scheme: Instead of simply computing ρ value by addition $\left(\rho = \sum_{i=1}^{d} \rho_i\right)$ we could use another Shamir's secret sharing and calculate ρ by interpolation

$$\rho = \sum_{i=1}^{d} \rho_i \prod_{j \in S j \neq i} \frac{x-j}{i-j}.$$

The new ciphertext would be

$$CT_{\mathcal{A}_\mathcal{C}} = (\mathcal{A}_\mathcal{C}, \rho_i T_{i,a}, \rho P + M)$$

In the decryption phase instead of multiply the $e(P, H(GID_u))^{\rho_k}$components we should interpolate them to derive the hiding factor of the message.

6.3 Comment 3

To improve our scheme we can apply the key policy ciphertext delegation to the scheme. The key policy ciphertext delegation was applied to attribute based encryption by Sahai et al. in [16]. They described the ciphertext delegation as follows: the ciphertext with a given access policy could be delegated to a more restrictive policy if there was a procedure that, given any valid encryption of a message under the first policy produces, is an independent and uniformly chosen encryption of the same message under the new access policy. We show a procedure for our scheme to delegate ciphertext.

For ciphertext,

$$CT_{\mathcal{A}_\mathcal{C}} = (\mathcal{A}_\mathcal{C}, \rho_i T_{i,a}, \rho P + M)$$

we can produce a new fresh encryption of the ciphertext

$$CT_{\mathcal{A}_{\mathcal{C}'}} = (\mathcal{A}_{\mathcal{C}'}, (\rho_i + \rho_{i'}) T_{i,a}, \rho' P + \rho P + M)$$

where $\rho' = \sum_{i=1}^{l} \rho_{i'}$ and $\mathcal{A}_\mathcal{C} \subseteq \mathcal{A}_{\mathcal{C}'}$. We also could add more authority to the protocol. We do it in 2 steps. At first, we need to update the ciphertext. The

updated ciphertext:

$$CT_{\mathcal{A}_{\mathcal{C}'}} = (\mathcal{A}'_{\mathcal{C}}, (\rho_i + \rho_{i'})T_{i,a}, \rho' P + \rho P + M)$$

To add a new authority $\mathcal{A}_j$ to the system, we choose a new $\rho_j \in_R \mathbb{Z}_p$ and compute $\rho_j T_{j,a}$ and add $\rho_j P$ to the ρP. The new encryption of the same message that includes a new authority $\mathcal{A}_j$:

$$CT_{\mathcal{A}_{\mathcal{C}'}} = (\mathcal{A}_{\mathcal{C}'}, (\rho_i + \rho_{i'})T_{i,a}, \rho_j T_{j,a}, \rho_j P + \rho' P + \rho P + M)$$

However we are unable to exclude any of the involved authorities without the help of the encryptor, but this is what we need from our scheme to be secure.

7 CONCLUSION

We constructed a multi-authority attribute based encryption scheme without any central authority. The authorities do not need to communicate with each other, not even be aware of each other. The security of the scheme depends on the decisional 3-party Diffie-Hellman assumption and it is proven to be secure in the selective-ID model. It has the advantage of KP-ABE schemes, an attribute authority is able to give an access structure that defines which user is able to decrypt under its authority name but it also gives the flexibility for the encryptor to be able to decide which authorities' attributes are used in the encryption (CP-ABE). It is still possible to improve the scheme by changing the used secret sharing scheme to more general secret sharing scheme, or by finding a better solution to find preimage of $e(M, H(GID_u))$ than using the lookup table. The size of the lookup table limits the size of the message space.

Acknowledgement

The research was carried out as part of the EITKIC_12-1-2012-0001 project, which is supported by the Hungarian Government, managed by the National Development Agency, financed by the Research and Technology Innovation Fund and it was performed in cooperation with the EIT ICT Labs Budapest Associate Partner Group (`www.ictlabs.elte.hu`).

REFERENCES

[1] Boneh, D.—Franklin, M. K.: Identity-Based Encryption from the Weil Pairing. CRYPTO, 2001, pp. 213–229.

[2] Bozovic, V.—Socek, D.—Steinwandt, R.—Villányi, V. I.: Multi-Authority Attribute Based Encryption with Honest-but-Curious Central Authority. Cryptology ePrint Archive Report 2009/083, February, 2009.

[3] Bozovic, V.—Socek, D.—Steinwandt, R.—Villányi, V. I.: Multi-Authority Attribute-Based Encryption with Honest-but-Curious Central Authority. International Journal of Computer Mathematics, Vol. 89, 2012, No. 3, pp. 268–283.

[4] Bethencourt, J.—Sahai, A.—Waters, B.: Ciphertext-Policy Attribute-Based Encryption. IEEE Symposium on Security and Privacy, 2007, pp. 321–334.

[5] Boneh, D.—Sahai, A.—Waters, B.: Functional Encryption Definitions and Challenges. TCC, 2011, pp. 253–273.

[6] Chase, M.—Chow, S. S. M.: Improving Privacy and Security in Multi-Authority Attribute-Based Encryption. Proceedings of the 16th ACM Conference on Computer and Communications Security, 2009, pp. 121–130.

[7] Chase, M.: Multi-Authority Attribute Based Encryption. TCC, 2007, pp. 515–534.

[8] Goyal, V.—Pandey, O.—Sahai, A.—Waters, B.: Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. Proceedings of the 13th ACM Conference on Computer and Communications Security, 2006, pp. 89–98.

[9] Lin, H.—Cao, Z.—Liang, X.—Shao, J.: Secure Threshold Multi Authority Attribute Based Encryption without a Central Authority. Information Sciences, Vol. 180, 2010, No. 13, pp. 2618–2632.

[10] Lewko, A. B.—Waters, B.: Unbounded HIBE and Attribute-Based Encryption. EUROCRYPT, 2011, pp. 547–567.

[11] Lewko, A. B.—Waters, B.: New Proof Methods for Attribute-Based Encryption Achieving Full Security through Selective Techniques. CRYPTO, 2012, pp. 180–198.

[12] Müller, S.—Katzenbeisser, S.—Eckert, C.: Distributed Attribute-Based Encryption. ICISC, 2008, pp. 20–36.

[13] Shamir, A.: How to Share a Secret. Communications of the ACM, Vol. 22, 1979, pp. 612–614.

[14] Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. CRYPTO '84, 1985, pp. 47–53.

[15] Sahai, A.—Waters, B.: Fuzzy Identity-Based Encryption. EUROCRYPT 2005, 2005, pp. 457–473.

[16] Sahai, A.—Seyalioglu, H.—Waters, B.: Dynamic Credentials and Ciphertext Delegation for Attribute-Based Encryption. CRYPTO, 2012, pp. 199–217.

Viktória I. Villányi is Assistant Professor at the Department of Operations Research of Eötvös Loránd University, Hungary. She holds her Ph.D. degree in mathematics from Florida Atlantic University, USA, 2009, and Ph.D. in applied informatics from University of Óbuda, Hungary, 2011. She received her Master of Science degree in mathematics education from Eötvös Loránd University, Hungary, in 2002.