Information Sciences 276 (2014) 354–362

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An expressive and provably secure Ciphertext-Policy Attribute-Based Encryption

A. Balu *, K. Kuppusamy

Department of Computer Science and Engineering, Alagappa University, Karaikudi, India

ARTICLE INFO

Article history:
Received 3 January 2012
Received in revised form 3 December 2013
Accepted 21 December 2013
Available online 31 December 2013

Keywords:
Attribute-Based Encryption
Linear integer secret sharing
Access control
Ciphertext-Policy Attribute-Based Encryption

ABSTRACT

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) allows to encrypt data under an access policy, specified as a logical combination of attributes. Such ciphertexts can be decrypted by anyone with a set of attributes that satisfy the access policy. We propose a Ciphertext-Policy Attribute-Based Encryption, which is based on a recent secret sharing method called Linear Integer Secret Sharing Scheme (LISS). In this scheme, the encryptor can specify the access policy in terms of LISS matrix *M*, over the attributes in the system. The scheme is selectively secure under Decisional Bilinear Diffie–Hellman (DBDH) assumption.

1. 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 many-to-many encryption instead of many-to-one. ABE is envisioned as an important tool for addressing the problem of secure and fine-grained data sharing and access control. In an ABE system, a user is identified by a set of attributes. In their seminal paper [16] they use biometric measurements as attributes in the following way. A secret key based on a set of attributes ω, can decrypt a ciphertext encrypted with a public key based on a set of attributes ω', only if the sets ω and ω' overlap sufficiently as determined by a threshold value *t*. 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 recruitment related document to all recruitment committee members in the Computer Science Department. In this case the document would be encrypted to the attribute subset “Faculty”, “CS Dept.”, “Recruitment Committee”, 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. There are two variants of ABE: Key-Policy based ABE (KP-ABE) [6] and Ciphertext-Policy based ABE (CP-ABE) [3]. In KP-ABE, the ciphertext is associated with a set of attributes and the secret key is associated with the access policy. The encryptor defines the set of descriptive attributes necessary to decrypt the ciphertext. The trusted authority, who generates user’s secret key, defines the combination of attributes for which the secret key can be used. In CP-ABE, the idea is reversed: now the ciphertext is associated with the access policy and the encrypting party determines the policy under which the data can be decrypted, while the secret key is associated with a set of attributes.

* Corresponding author. Tel.: +91 9443841911.
E-mail addresses: balusuriya@yahoo.co.in (A. Balu), kkdiksamy@yahoo.com (K. Kuppusamy).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.12.027

1.1. Motivation

Up to date, in most of CP-ABE schemes, the secret exponent s is shared by Shamir's secret sharing scheme. Recently, Waters [19] proposed three CP-ABE schemes, which are based on Linear Secret Sharing Scheme (LSSS). In 2006, Damgard and Thorbek [5] introduced the notion of Linear Integer Secret Sharing (LISS) scheme, and showed that the construction of their scheme is suitable for any access structure. Access structure is represented using the Boolean operators AND, OR. In the access structure, the variable x_i can appear more than once and there is no bound for the occurrence of the variable, where as in [19] there is a bound for the occurrence of a variable x_i. In LISS, the focus has been on the advantages of secret sharing over integers opposed to secret sharing over finite groups or fields in LSSS. In LISS, the secret sharing cost is less. Using these advantages, it is very easy to express the access policy effectively and share the secret exponent s efficiently in our CP-ABE construction.

1.2. Our contribution

We present a new scheme for constructing a Ciphertext-Policy ABE based on Linear Integer Secret Sharing Scheme (LISS), which allows to represent any attribute access policy by a distribution matrix M. Access policy can be expressed more effectively and efficiently by this scheme. In this scheme, the ciphertext is associated with a $d \times e$ distribution matrix M. The matrix M can be formulated by using three rules to represent the attributes present in the access policy. In our method, we allow the repetition of the same attributes in the access policy, where as in Water's [19] method they have an exclusive method for the repetition of the attributes. The secret$s$ can be selected from the publically known interval $[-2^\ell, 2^\ell]$. We use ρ as distribution vector and the secret can be shared by the components of $M \cdot \rho$. Any one who satisfies the access policy, is able to decrypt the ciphertext. We provide a solution that is secure under the Decisional Bilinear Diffie–Hellman assumption.

1.3. Related work

Sahai and Waters [16] have introduced Attribute-Based Encryption (ABE). Bethencourt et al. [3] has proposed the first Ciphertext-Policy ABE using threshold secret sharing to enforce the policy in the encryption phase. This method requires polynomial interpolation to reconstruct the secret and secure in the generic group model. The CP-ABE proposed by Cheung and Newport [4], in which decryption policies are restricted to a single AND gate, and attributes are allowed to be either positive or negative. In this method, the size of the ciphertext and secret key increases linearly with the total number of attributes in the system. Goyal et al. [7] have given a "bounded" CP-ABE construction based on the tree-based access structure. The disadvantage of their scheme is that the depth of the access trees d under which messages can be encrypted is defined in the setup phase. Thus, the user who wants to encrypt a message is restricted to use only an access tree which has the depth $d' \leqslant d$. Water's [19] presented three constructions, which are based Linear Secret Sharing Scheme (LSSS) and secure under various difficulty assumption. Ciphertext and public parameters size have been increased in the DBDH assumption [19]. Ibraimi et al. [8] proposed a CP-ABE scheme in which the secret$s$ can be split by Shamir's Secret Sharing scheme or by Unanimous consent control by modular addition scheme. If the access policy contains OR and *of* operators then, the whole secret s will be assigned to the attributes. Most of the ABE constructions have been proved to be selectively secure. Lewko et al. [9] first obtain full security by adapting the dual system encryption technique [18] to the ABE case. Okamoto and Takashima [14] present a fully secure ABE scheme under a well-established assumption. The scheme in [10] is fully secure as well.

2. Preliminaries

2.1. Access structures

Definition 1 (*Access structure*). Let $\{1, 2, \ldots n\}$ be a set of parties. A collection $\Gamma \subseteq 2^{\{1,2,\ldots,n\}}$ is monotone if $\forall B, C:$ if $B \in \Gamma$ and $B \subseteq C$ then $C \in \Gamma$. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) A of non-empty subsets of $\{1, 2 \ldots, n\}$ i.e., $\Gamma \subseteq 2^{\{1,2,\ldots,n\}} \setminus \phi$. The sets in Γ are called the authorized sets, and the sets not in Γ are called the unauthorized sets.

2.2. Linear integer secret sharing

In the LISS scheme, the secret is an integer chosen from a (publicly known) interval, and each share is computed as an integer linear combination of the secret and some random numbers chosen by the dealer. Reconstruction of the secret is done by computing a linear combination with integer coefficients of the shares in a qualified set. Let $P = \{1, 2, \ldots, n\}$ denote the n share holders and D the dealer. Let Γ be a monotone access structure on P. Let ℓ be an integer constant. The dealer D wants to share a secret s from the publicly known interval $\left[-2^\ell, 2^\ell\right]$ to the shareholders P over Γ, such that every set of shareholders $A \in \Gamma$ can reconstruct s, but a set of shareholders $A \notin \Gamma$ get no or little information on s.

We say that a subset $A \subseteq P$ is qualified if the parties in A jointly are allowed to reconstruct the secret s. In a LISS scheme, the shares consist of a collection of integers,$\{s_i\}_{i\in I}$, where for each $i \in I$, the integer s_i belongs to exactly one party and s_i is computed by a linear integer combination of $s$and some randomness chosen by the dealer. Given a qualified subset of shares $\{s_i\}_{i\in I'}$, then the secret can be reconstructed by a linear combination $s = \sum_{i\in I'} \lambda_i s_i$, where $\{\lambda_i\}_{i\in I'}$ are integer coefficients that are determined by the index I'. We use a distribution matrix $M \in Z^{d\times e}$ and a corresponding surjective function $\Psi : \{1, \ldots, d\} \rightarrow P$. We say that the ith row is labeled by $\Psi(i)$ or owned by party $P_{\Psi(i)}$. We use a distribution vector $\rho = (s, \rho_2, \ldots, \rho_e)$ where s is the secret, and the $\rho_i' s$ are uniformly random chosen integers in $[-2^{\ell_0+k}, 2^{\ell_0+k}]$, where k is the security parameter and ℓ_0 is a constant. The dealer D calculates shares by

$$M \cdot \rho = (s_1, \ldots, s_d)^T \tag{1}$$

where we denote each s_i as a share unit for $1 \leqslant i \leqslant d$ and$T$ denotes the transpose. The ith share unit is then given to the $\Psi(i)'$th shareholder. If $A \subseteq P$ is a set of shareholders, then M_A denotes the restriction of M rows jointly owned by A.

Definition 2. A LISS scheme is correct, if the secret is reconstructed from shares $\{s_i / i \in A\}$ where A is a authorized set of shareholders, by taking an integer linear combination of the shares, with coefficient that depend only on the index set A.

Definition 3. A LISS scheme is private, if for any two secrets $s, s' \in [-2^{\ell}, 2^{\ell}]$, and independent random coins r, r' and any unauthorized set A of shareholders, the distribution of $\{s_i(s, r, k) / i \in A\}$ and $\{s_i(s', r', k) / i \in A\}$ are statistically indistinguishable. More precisely, the statistical distance between the two distributions is negligible in k.

2.3. Integer span program

Definition 4. $\mathcal{M} = (M, \Psi, \xi)$ is called an Integer Span Program (ISP) if $M \in Z^{d\times e}$ and the d rows of M are labeled by a surjective function $\Psi : \{1, \ldots, d\} \rightarrow P$. Finally $\xi = (1, 0, 0, \ldots 0)^T \in Z^e$ is called the target vector. We define $size(\mathcal{M}) = d$, where d is the number of rows of M.

Definition 5. Let Γ be a monotone access and let $\mathcal{M} = (M, \Psi, \xi)$ be an integer span program. Then $\mathcal{M}$ is an ISP, if for all $A \subseteq \{1, \ldots, n\}$ the following holds.

1. If $A \in \Gamma$, then there is a vector $\lambda \in Z^d$ such that $M_A^T \lambda = \xi$.
2. If $A \notin \Gamma$, then there exists $\mathbf{k} = (k_1, \ldots, k_e)^T \in Z^e$ such that $M_A \cdot \mathbf{k} = 0 \in Z^d$ with $k_1 = 1$, which is called the sweeping vector for A.

If we have an ISP $\mathcal{M} = (M, \Psi, \xi)$ which computes Γ, we build a LISS scheme for Γ as follows: We use M as the distribution matrix and $\ell_0 = \ell + \lceil \log_2(k_{max}(e-1)) \rceil + 1$, where ℓ is the length of the secret and $k_{max} = max\{|a| / a$ is an entry in some sweeping vector$\}$.

Lemma. *If $A \in \Gamma$, then there exists a reconstruction vector λ_A such that $M_A^T \lambda_A = \xi$.*

Proof. We can reconstruct λ_A based on the induction in the number of variables (attributes) present in the formula $\mathcal{P}$ that represents the access structure Γ.

If $M \in Z^{d\times e}$, then we construct the reconstruction vector $\lambda \in Z^d$ for $A \in \Gamma$, such that $M^T \cdot \lambda = \xi$.

In the initial case, $\mathcal{P} = x$, there is only one party, and the distribution matrix is $M = (1)$, i.e., the party gets the secret s, so the reconstruction vector is $\lambda = (1)^T$.

In the case of an OR-term, $\mathcal{P} = \mathcal{P}_a \bigvee \mathcal{P}_b$, then M_a and M_b be the matrices that represent the formulas $\mathcal{P}_a$ and $\mathcal{P}_b$ respectively. It is obvious that one of the matrices is enough to reconstruct the secret. Take the reconstruction vector belonging to the matrix with which it is possible to reconstruct the secret, e.g., $\lambda_a = (\lambda_1, \ldots, \lambda_t)^T$. The new reconstruction vector for the OR-term is $\lambda_{OR} = (\lambda_1, \ldots, \lambda t, 0, \ldots, 0)$, we put zeros in the entries that represent the shares from M_b.

In the case where there is an AND-term $\mathcal{P} = \mathcal{P}_a \bigwedge \mathcal{P}_b$, then let the matrices M_a and M_b represent the formulas $\mathcal{P}_a$ and $\mathcal{P}_b$ respectively. By assumption each of the matrices M_a and M_b can reconstruct their part of the secret, let $\lambda_a = (\lambda_{a_1}, \ldots, \lambda_{a_t})^T$ and $\lambda_b = (\lambda_{b_1}, \ldots, \lambda_{b_t})^T$ be the reconstruction vector for the AND-term is:

$\lambda_{AND} = (\lambda_{a_1}, \ldots, \lambda_{a_t}, -\lambda_{b_1}, \ldots, -\lambda_{b_t})^T$, because, if we define

$$\lambda_{a'} = (\lambda_{a_1}, \ldots, \lambda_{a_t}, 0, \ldots, 0)^T$$
$$\lambda_{b'} = (0, \ldots, 0, \lambda_{b_1}, \ldots, \lambda_{b_t})^T$$

we know that if M is the distribution matrix that represents the AND-term, then

$$(M \cdot \rho)^T \cdot \lambda_{a'} = s + \rho_2$$
$$(M \cdot \rho)^T \cdot \lambda_{b'} = \rho_2$$

i.e., we have that

$$(M \cdot \rho)^T \cdot \lambda_{a'} - (M \cdot \rho)^T \cdot \lambda_{b'} = (M \cdot \rho)^T \cdot (\lambda_{a'} - \lambda_{b'}) = (M \cdot \rho)^T \cdot \lambda_{AND} = s + \rho_2 - \rho_2 = s$$

which concludes the proof. □

2.3.1. Secret reconstruction

An authorized set A can compute the secret by taking a linear combination of their values, since there exists $\lambda_A \in Z^{d_A}$ such that $M_A^T \cdot \lambda_A = \xi$ (as per Definition 5).

With the secret shares of s_A it is justified to reconstruct the secret$s$ by the following way

$$\begin{aligned} s_A^T \cdot \lambda_A &= (M_A \cdot \rho)^T \cdot \lambda_A \text{ (by Eq. (1))} \\ &= \rho^T \cdot (M_A^T \cdot \lambda_A) = \rho^T \cdot \xi = s \end{aligned}$$

2.4. Bilinear maps

Let G_0, G_1, be two multiplicative cyclic groups of prime order p. Let g be a generator of G_0. Let e be a bilinear map, $e: G_0 \times G_0 \rightarrow G_1$. The bilinear map e has the following properties:

1. Bilinearity: for all $u, v \in G_0$, and $a, b \in \mathbb{Z}_p^*$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$.

The map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.5. Decisional Bilinear Diffie–Hellman assumption

We define the Decisional Bilinear Diffie–Hellman problem as follows. A challenger chooses a group G_0 is of prime order p according to the security parameter. Let $a, b, s \in \mathbb{Z}_p^*$ be chosen at random and g be a generator G_0. The adversary given (g, g^a, g^b, g^s) must distinguish a valid tuple $e(g, g)^{abs} \in G_1$ from a random element Rin G_1. An algorithm $\mathcal{A}$ that outputs $\{0,1\}$ has advantage ϵ in solving decisional BDH in G_0 if

$$Pr[\mathcal{A}(g, g^a, g^b, g^s, D = e(g, g)^{abs})) = 0] - Pr[\mathcal{A}(g, g^a, g^b, g^s, D = R) = 0] \geqslant \epsilon$$

where Pr denotes the probability.

2.6. Ciphertext-Policy Attribute-Based Encryption

A Ciphertext-Policy Attribute-Based Encryption scheme consists of four fundamental algorithms: Setup, Key Generation, Encryption and Decryption. Let U be the set of attributes.

2.6.1. Setup

The setup algorithm takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK.

2.6.2. KeyGen (MK, S)

The key generation algorithm takes as input the master key MK and a set of attributes S. It outputs a private key SK for the attributes in S.

2.6.3. Encrypt (PK, $\mathcal{P}$, m)

The encryption algorithm takes as input the public parameters PK, the message m, and an access structure $\mathcal{P}$ over the universe of attributes. The algorithm will encrypt m and produce a ciphertext CT such that only a user that possesses a set of attributes that satisfy the access structure will be able to decrypt the message. Assume that the ciphertext implicitly contains $\mathcal{P}$.

2.6.4. Decrypt (CT,SK)

The decryption algorithm takes as input the ciphertext CT, which contains an access structure $\mathcal{P}$, and a private key SK, which is a private key for a set S of attributes. If the set S of attributes satisfies the access structure $\mathcal{P}$ then the algorithm will decrypt the ciphertext and return a message m.

2.7. Security model for CP-ABE

The semantic security against chosen-plaintext attack (CPA) is modeled in the selective attribute model (sAtt), where the adversary must provide the challenge access tree he wishes to attack before he receives the public parameters from the challenger. The game is carried out between a challenger and an adversary. Specifically, the game is as follows.

2.7.1. Init

The adversary chooses the challenge access policy τ^* and gives it to the challenger.

2.7.2. Setup

The challenger runs the Setup algorithm and gives the public parameters, PK to the adversary.

2.7.3. Phase1

The adversary makes a secret key request to the KeyGen oracle for any attribute set $\omega = \{a_j/a_j \in U\}$ with the restriction that ω not satisfying τ^*. The Challenger returns KeyGen (MK,ω).

2.7.4. Challenge

The adversary submits two equal length messages M_0 and M_1. The Challenger flips a random coin d, and encrypts M_d under τ^*. The ciphertext CT^* is given to the adversary.

2.7.5. Phase 2

The adversary can continue querying KeyGen with the same restriction as during Phase1.

2.7.6. Guess

The adversary outputs a guess d' of d.

Definition 6. A ciphertext-policy attribute based encryption scheme is said to be secure against a chosen-plaintext attack (CPA) in the selective attribute model if any polynomial time adversaries have only a negligible advantage in the IND-sAtt-CPA game, where the advantage is defined to be $\epsilon = |Pr[d' = d] - \frac{1}{2}|$.

3. Main construction

In our CP-ABE construction, it is required to convert the access policy into a distribution matrix M. The matrix M can be formulated using the following three rules [5]. After constructing the distribution matrix M, shares of the attributes are calculated using the distribution matrix M. Message m will be encrypted and then the attributes present in the access policy are encrypted using the corresponding attribute shares. Any one who satisfies the access policy is able to decrypt the ciphertext.

Now, we specify the method to form the access policy matrix M and then the construction of the encryption scheme.

3.1. Formation of access policy matrix M

Let $M_u \in Z^{1\times 1}$ be the matrix with single entry which is one, i.e., $M_u = [1]$. If we have a matrix $M_a \in Z^{d_a\times e_a}$ then we can form $c_a \in Z^e$ to represent the first column in M_a and $R_a \in Z^{(d_a-1)\times e_a}$ to represent all but the first column in M_a.

3.1.1. Rule 1

Each variable a_i in the access policy $\mathcal{P}$ can be expressed by M_u.

3.1.2. Rule 2

For any OR-term $\mathcal{P} = \mathcal{P}_a \bigvee \mathcal{P}_b$. Let $M_a \in Z^{d_a\times e_a}$ and $M_b \in Z^{d_b\times e_b}$ be the matrices which express the formulas $\mathcal{P}_a$ and $\mathcal{P}_b$ respectively. We can construct a matrix $M_{OR} \in Z^{(d_a+d_b)(e_a+e_b-1)}$ expressing $\mathcal{P}$, which is defined by letting the first column of M_{OR} be the concatenation of the two column vectors c_a and c_b, then letting the following $d_a - 1$ columns be the columns of R_a expanded with e_b succeeding zero entries, and the last $d_b - 1$ columns be the columns of R_b expanded with e_a leading zero entries. This is visualized by

$$M_{OR} = \begin{array}{ccc} c_a & R_a & 0 \\ c_b & 0 & R_b \end{array}$$

3.1.3. Rule 3

For any AND-term $\mathcal{P} = \mathcal{P}_a \bigwedge \mathcal{P}_b$. Let $M_a \in Z^{d_a\times e_a}$ and $M_b \in Z^{d_b\times e_b}$ be the matrices which express the formulas $\mathcal{P}_a$ and $\mathcal{P}_b$ respectively. We can construct a matrix $M_{AND} \in Z^{(d_a+d_b)(e_a+e_b)}$ which expresses the access policy $\mathcal{P}$. It is defined by letting the first column of M_{AND} be the column vector c_a expanded with e_b succeeding zero entries, the next column to be the concatenation of c_a and c_b the following $d_a - 1$ columns be the columns of R_a expanded with e_b succeeding zero entries, and the last $d_b - 1$ columns be the columns of R_b expanded with e_a leading zero entries. This is visualized by

$$M_{AND} = \begin{matrix} c_a & c_a & R_a & 0 \\ 0 & c_b & 0 & R_b \end{matrix}$$

For example if we have an access policy $\mathcal{P} = (a_1 \bigwedge a_2) \bigvee (a_3 \bigwedge a_4)$, then $\mathcal{P}_a = (a_1 \bigwedge a_2),\ \mathcal{P}_b = (a_3 \bigwedge a_4)$.

$\mathcal{P}_a$ can be represented as $M_a = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ and $\mathcal{P}_b$ can be represented as $M_b = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ by using AND-rule on a_1 and a_2, a_3 and a_4 respectively. We can express$M$ by using OR-rule on $\mathcal{P}_a$, $\mathcal{P}_b$ and their respective matrix M_a, M_b finally.

$$M = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

Note that each attribute in the access policy owns the corresponding row in the resulting matrix M.

If an attribute presents more than once in the access policy, then it owns more than one row in the resulting matrix M. For example if we have an access policy $\mathcal{K} = (a_1 \bigwedge a_2) \bigvee (a_1 \bigwedge a_3) \bigvee (a_2 \bigwedge a_3)$, then the matrix$M$is given by

$$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where rows 1 and 3 are owned by attribute a_1, rows 2 and 5 are owned by a_2, and rows 4 and 6 are owned by attribute a_3 finally. Also, the above access policy $\mathcal{K}$ represents the threshold 2 out of 3 access structure.

In the above example the minimal sets which can reconstruct the secret are:

$$(a_1, a_2),\ (a_1, a_3),\ \text{and}\ (a_2, a_3).$$

If the party possessing the attributes (a_1, a_2) want to reconstruct the secret, they need to use the reconstruction vector given by $\lambda = (1, -1, 0, 0, 0, 0)^T$.

On the other hand, if the parties possessing the attributes (a_2, a_3) want to reconstruct the secret, they need to use the reconstruction vector given by $\lambda = (0, 0, 0, 0, 1, -1)^T$.

3.2. CP-ABE scheme

3.2.1. Setup (1^k)

The setup algorithm chooses a group G_0 of prime order$p$ and a generator g. Let $U = \{a_1, a_2, \ldots, a_n\}$ be the set of attributes. It chooses random elements $t_1, t_2, \ldots, t_n,\ \alpha \in Z_p$. Let $y = e(g, g)^{\alpha}$, and $T_j = g^{t_j}(1 \leqslant j \leqslant n)$.

The Public Key is $\mathrm{PK} = (g, y, T_j(1 \leqslant j \leqslant n))$ and the Master Secret Key is $\mathrm{MK} = (\alpha, t_j(1 \leqslant j \leqslant n))$.

3.2.2. KeyGen (MK, S)

This algorithm takes as input the master secret key and a set S of attributes and performs the following:

(a) Select random values $a, r \in Z_p$ and compute $d_0 = g^{\alpha - ar}$.
(b) For each attribute $a_j \in S$, compute $d_j = g^{art_j^{-1}}$.
(c) The secret key is $\mathrm{SK} = (d_0, \forall a_j \in S : d_j)$.

3.2.3. Encrypt (PK, $\mathcal{P}$, m)

The encryption algorithm takes as input the public key, a message $m \in G_1$ to encrypt and the access policy $\mathcal{P}$.

Step 1 Select a random element $s \in [-2^{\ell}, 2^{\ell}]$ and compute $C_0 = g^s$. M is the distribution matrix constructed by the above method for the access policy $\mathcal{P}$. Choose $\rho = (s, \rho_2, \ldots, \rho_e)^T$, where $\rho_i's$ are uniformly random chosen integers in $[-2^{\ell_0+k}, 2^{\ell_0+k}]$.
Step 2 Compute $M \cdot \rho = (s_1, \ldots, s_d)^T$.
(a) $C' = m \cdot y^s = m \cdot e(g, g)^{\alpha s}$.
(b) For each attribute in $\mathcal{P}$, compute $C_i = T_i^{s_i}$ using the corresponding shares of the attribute a_i.

The ciphertext is published as $CT = (C_0, C', C_i; i = 1 \text{ to } d)$ along with M.

3.2.4. Decrypt (CT, SK)

The decryption algorithm takes as input a ciphertext CT along with M and a private key for a set $A \subseteq S$. Suppose A satisfies the access policy $\mathcal{P}$, then there is a vector $\lambda_A \in Z^{d_A}$ such that $M_A^T \lambda_A = \xi$ (as per Definition 5). With this, it is possible to reconstruct the secret using $\sum_{i \in A} \lambda_i s_i = s$.

The decryption algorithm computes

$$\begin{aligned}
&\frac{C'}{\left(e(C_0, d_0)\prod_{i \in A} e(C_i, (d_i)^{\lambda_i})\right)} \\
&= \frac{m \cdot e(g,g)^{\alpha s}}{\left(e(g^s, g^{\alpha - ar})\prod_{i \in A} e\left(T_i^{s_i}, \left(g^{art_i^{-1}}\right)^{\lambda_i}\right)\right)} \\
&= \frac{m \cdot e(g,g)^{\alpha s}}{\left(e(g^s, g^{\alpha - ar})\prod_{i \in A} e\left(g^{t_i s_i}, \left(g^{art_i^{-1}}\right)^{\lambda_i}\right)\right)} \\
&= \frac{m \cdot e(g,g)^{\alpha s}}{\left(e(g^s, g^{\alpha - ar})\prod_{i \in A} e(g,g)^{ars_i \lambda_i}\right)} \\
&= \frac{m \cdot e(g,g)^{\alpha s}}{(e(g^s, g^{\alpha - ar})\prod_{i \in A} e(g,g)^{ars})} \\
&= \frac{m \cdot e(g,g)^{\alpha s}}{(e(g^s, g^{\alpha - ar}) e(g,g)^{ars})} \\
&= m
\end{aligned}$$

3.3. Security analysis

3.3.1. Theorem 1

Suppose the Decisional Bilinear Diffie–Hellman assumption holds, then no polynomial adversary can selectively break our system.

3.3.2. Proof

Suppose we have an adversary $\mathcal{A}$ with non-negligible advantage ϵ in the selective security game against our construction. We show how to use the adversary $\mathcal{A}$ to build a simulator $\mathcal{B}$ that is able to solve the DBDH assumption. The Challenger gives the simulator $\mathcal{B}$ the DBDH challenge: $(g, A, B, C, D) = (g, g^a, g^b, g^s, D)$.

3.3.3. Init

The adversary chooses the challenge access policy (M', p^*) and gives it to the simulator.

3.3.4. Setup

The simulator selects at random $a' \in Z_p$ and implicitly sets $\alpha = ab + a'$ by letting $e(g,g)^{\alpha} = e(g^a, g^b)e(g,g)^{a'}$. For all $a_j \in U$, it chooses a random $q_j \in Z_p$ and set $T_j = g^{\frac{1}{\left(M'_{i,j} q_j\right)}}$ if $a_j \notin p^*$, otherwise $T_j = g^{q_j}$.

The simulator $\mathcal{B}$ sends the public parameters to $\mathcal{A}$.

3.3.5. Phase 1

$\mathcal{A}$ makes secret key requests for any set of attributes $\omega = \{a_j / a_j \in U\}$ with the restriction that $a_j \nvDash p^*$. On each request $\mathcal{B}$ chooses a random variable $v \in Z_p$, and finds a vector $\mathbf{k} = (k_1, k_2, \ldots, k_e)^T \in Z^e$ such that $M' \cdot \mathbf{k} = \mathbf{0}$ with $k_1 = 1$. By the definition of Sweeping vector, such a vector must exist. Simulator sets r value as $v + k_j b$.

Choose k_j as k_1 to compute,

$$d_0 = g^{\alpha - a(v + k_1 b)} = g^{ab + a' - av - ab} = g^{a'} A^{-v}$$

In calculating d_j, we have the term $M'_{i,j} a \cdot k_j b$ and it gets canceled because of $M' \cdot \mathbf{k} = \mathbf{0}$

$$d_j = g^{a(v + k_j b) q_j M'_{i,j}} = A^{v M'_{i,j} q_j}$$

$$d_0 = g^{a'} A^{-v}, \quad d_j = A^{v M'_{i,j} q_j}, \quad \forall a_j \in \omega$$

Table 1
Comparison of schemes.

Method	CT	PKS	EN	DE
GJPS [7]	$\Theta(U.n_{max}^{3.42})$	$\Theta(A.n_{max}^{3.42})$	$\Theta(U.n_{max}^{3.42})$	$\Theta(U.n_{max}^{3.42})$
Waters [19]	$\Theta(n^2)$	$\Theta(k_{max}.A + n_{max})$	$\Theta(n^2)$	$\Theta(n.T)$
Our method	$\Theta(n)$	$\Theta(A)$	$\Theta(n)$	$\Theta(T)$

3.3.6. Challenge

$\mathcal{A}$ submits two messages $m_0, m_1 \in G_1$. The simulator flips a fair binary coin d, and returns the encryption of m_d. The encryption of m_d can be done as follows:

$$C_0 = g^s, \quad C' = m_d De(g^s, g^{a'})$$

The simulator will choose uniformly random integers $z_2, \ldots, z_h$ in $[-2^{\ell_0+k}, 2^{\ell_0+k}]$ and share the secret s using the vector $\Phi = (s, z_2, \ldots, z_h)$.

Create the distribution matrix M, for the access policy p^*. Compute $M \cdot \Phi$ and use the shares to encrypt the access policy with corresponding q_j for the attributes present in the access policy p^*, $C_j = T_j^{S_j}$.

3.3.7. Phase 2

Same as Phase 1.

3.3.8. Guess

$\mathcal{A}$ outputs a guess d' of d. The simulator then outputs 0 to the guesses that $D = e(g,g)^{abs}$ if $d' = d$; otherwise, it outputs 1 to indicate that it believes D is random group element in G_1.

When D is a tuple, the simulator $\mathcal{B}$ gives a perfect simulation. So we have that $Pr[\mathcal{B}(\rho, D = e(g,g)^{abs}) = 0] = \frac{1}{2} + \epsilon$.

When D is a random group element, the message m_d is completely hidden from the adversary and we have $Pr[\mathcal{B}(\rho, D = R) = 0] = \frac{1}{2}$.

4. Efficiency

In Table 1, we give the comparison with Goyal et al. [7], Waters [19] and our method in terms of Ciphertext size (CT), Private Key Size (PKS), Encryption time (EN), Decryption time (DE) based on DBDH assumption. Let n be the number of attributes present in the access policy, A be the number of attributes in user's key, T be the number of nodes satisfied by a user's attributes, U be the number of attributes defined in the system, nmax be the bound on the size of the access formula, kmax be the maximum number of times a single attribute will appear in a particular formula. Our CP-ABE method achieves significantly better performance than Waters [19] and GJPS [7] method.

5. Conclusion

We proposed a new type of Ciphertext-Policy Attribute-Based Encryption based on linear integer secret sharing scheme. This scheme is very expressive and provably secure under the Decisional Bilinear Diffie–Hellman assumption.

References

[3] J. Bethencourt, A. Sahai, B. Waters, Ciphertext policy attribute based encryption, in: IEEE Symposium on Security and privacy, 2007, pp. 321–334.

[4] L. Cheung, C. Newport, Provably secure ciphertext policy ABE, in: Proceedings of the 14th ACM Conference on Computer and Communications security CCS, 2007, pp. 456–465.

[5] I. Damgard, R. Thorbek, Linear integer secret sharing and distributed exponentiation, in: PKC, 2006, pp. 75–90.

[6] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine grained access control of encrypted data, in: ACM Conference on Computer and Communication Security, 2006, pp. 89–98.

[7] V. Goyal, A. Jain, O. Pandey, A. Sahai, Bounded ciphertext policy attribute based encryption, in: L. Aceto, I. Damgard, L.A. Goldberg, M.M. Halldorsson, A. INgolfsdottir, I. Walukiewicz (Eds.), ICALP, 2008, pp 579–591.

[8] L. Ibraimi, Q. Tang, P. Hartel, W. Jonker, Efficient and provable secure ciphertext-policy attribute based encryption schemes, in: F. Bao, H. Li, G. Wang (Eds.), ISPEC, 2009, pp. 1–12.

[9] A. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters. Fully secure functional encryption: attribute-based encryption and (Hierarchical) inner product encryption, In: H. Gilber (Ed.), EUROCRYPT, 2010, pp 62–91.

[10] A. Lewko, B. Waters, Decentralizing attribute-based encryption, in: EUROCRYPT, 2011, pp. 568–588.

[14] T. Okamoto, K. Takashima, Fully secure functional encryption with general relations from the decisional linear assumption, in: T. Rabin (Ed.), CRYPTO, 2010, pp. 191–208.

[16] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: R. Cramer (Ed.), EUROCRYPT, 2005, pp. 457–473.

[18] B. Waters, Dual system encryption: realizing fully secure IBE and HIBE under simple assumption, in: S. Halevi (Ed.), CRYPTO, 2009, pp. 153–170.

[19] B. Waters, Ciphertext policy attribute based encryption: an expressive, efficient, and provably secure realization, in: PKC, 2011, pp. 53–70.

Further Reading

[1] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. Panafieu, C. Rafols, Attribute-based encryption schemes with constant-size ciphertexts, Theor. Comput. Sci. (2012) 15–38.
[2] A. Beimel, Secure Schemes for Secret Sharing and Key Distribution, Ph.D Thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
[11] A. Lewko, B. Waters, New proof methods for attribute-based encryption: achieving full security through selective techniques, in: CRYPTO, 2012, pp 180–198.
[12] S. Liu, Y. Long, K. Chen, Key updating technique in identity-based encryption, Inform. Sci. 181 (2011) 2436–2440.
[13] H.C. Lu, H.L. Fu, New bounds on the average information rate of secret-sharing schemes for graph-based weighted threshold access structures, Inform. Sci. 240 (2013) 83–94.
[15] B. Qin, Q.H. Wu, L. Zhang, O. Farras, J. Doming-Ferrer, Provably secure threshold public key encryption with adaptive security and short ciphertexts, Inform. Sci. 200 (2012) 67–80.
[17] L. Wang, J. Shao, Z. Cao, M. Mambo, A. Yamamura, L. Wang, Certifcate-based proxy decryption systems with revocability in the standard model, Inform. Sci. 247 (2013) 188–201.
[20] Z. Zhang, L. Zhu, L. Liao, M. Wang, Computationally sound symbolic security reduction analysis of the group key exchange protocols using bi-linear pairings, Inform. Sci. 209 (2012) 93–112.