

# An Efficient and Expressive Ciphertext-Policy Attribute-Based Encryption Scheme with Partially Hidden Access Structures

Hui Cui<sup>1</sup>(✉), Robert H. Deng<sup>1</sup>, Guowei Wu<sup>1</sup>, and Junzuo Lai<sup>2</sup>

<sup>1</sup> School of Information Systems,  
Singapore Management University, Singapore, Singapore  
{hcui,robertdeng,gwu}@smu.edu.sg

<sup>2</sup> Department of Computer Science, Jinan University, Guangzhou, China  
pddlaijunzuo@163.com

**Abstract.** A promising solution to protect data privacy in cloud storage services is known as ciphertext-policy attribute-based encryption (CP-ABE). However, in a traditional CP-ABE scheme, a ciphertext is bound with an explicit access structure, which may leak private information about the underlying plaintext in that anyone having access to the ciphertexts can tell the attributes of the privileged recipients by looking at the access structures. A notion called CP-ABE with partially hidden access structures [14, 15, 18, 19, 24] was put forth to address this problem, in which each attribute consists of an attribute name and an attribute value and the specific attribute values of an access structure are hidden in the ciphertext. However, previous CP-ABE schemes with partially hidden access structures only support access structures in AND gates, whereas a few other schemes supporting expressive access structures are computationally inefficient since they are built from bilinear pairings over the composite-order groups. In this paper, we focus on addressing this problem, and present an expressive CP-ABE scheme with partially hidden access structures in prime-order groups.

**Keywords:** Cloud storage · Ciphertext-policy attribute-based encryption · Access structures · Data privacy · Access control

## 1 Introduction

With the explosive growth of information, there is an increasing demand for outsourcing data to cloud storage services due to its economical scale. However, no user would like to store documents containing sensitive information to a public cloud with no guarantee for security or privacy. A promising solution to provide data privacy while sharing data in cloud is using an encryption mechanism such that data owners upload their data in encrypted forms to the cloud and share them with users having the required credentials (or attributes). One encryption technique that meets this requirement is called ciphertext-policy attribute-based

**Table 1.** Comparisons of CP-ABE schemes with partially hidden access structures

Schemes	Anonymity of hidden access structures	Expressiveness of access structures	Type of bilinear group	Security	Unbounded attribute names
[19]	partially hidden	AND gates	prime	selective	no
[18]	partially hidden	AND gates	prime	selective	yes
[14]	partially hidden	AND gates	composite	full	no
[15]	partially hidden	LSSS	composite	full	no
[24]	partially hidden	AND gates	prime	selective	yes
Our scheme	partially hidden	LSSS	prime	selective	yes

encryption (CP-ABE) [3], in which a user’s private key issued by an attribute authority (AA) is associated with a set of attributes, a message is encrypted under an access structure (or access policy) over a set of attributes by the data owner, and a user can decrypt the ciphertext using his/her private key if and only if his/her attributes satisfy the access policy ascribed to this ciphertext.

Though a ciphertext in a traditional CP-ABE scheme (e.g., [3, 7, 16, 23]) does not directly tell the identities of its recipients, an access structure in the cleartext is attached to the ciphertext, and thus anyone who sees a ciphertext may be able to deduce certain private information about the encrypted message or the privileged recipients of the message. Let us consider the cloud storage system, which is used by a hospital to store electrical medical records (EMRs) of patients. In this system, the hospital encrypts an EMR using CP-ABE under an access structure “(Patient: NR005289 AND Hospital: City Hospital) OR (Doctor: Cardiologist AND Hospital: General Hospital)”, and then uploads the ciphertext together with the access policy to the cloud. The access policy requires that a patient identified by NR005289 at City Hospital or any Cardiologist at General Hospital can decrypt the ciphertext to obtain the EMR, from which it can be easily inferred that a person in City Hospital with a patient number NR005289 is suffering a heart problem. This information leakage is definitely not expected by the cloud users, and thus it is necessary to design CP-ABE schemes that can hide access structures.

It is known from [15] that a CP-ABE scheme with hidden access structures can be built from attribute-hiding Inner-product Predicate Encryption (IPE) [13], but this will result in an increase in the size for an arbitrary access structure in the transformation. Also, it is inefficient to implement CP-ABE schemes with fully hidden access structure from attribute-hiding IPE [16]. With the goal of having a trade off between fully hidden access structures and efficiency of CP-ABE, partially hidden access structures [14, 15, 18, 19, 24] were embedded in CP-ABE schemes to mitigate the computational cost. However, the schemes in [14, 18, 19, 24] can only be applied to access structures expressed in AND gates. The construction in [15] supports expressive access structures but is built from pairings over the composite-order groups, and “a Tate pairing on a 1024-bit composite-order elliptic curve is roughly 50 times slower than the same pairing on a comparable prime-order curve,

and this performance gap will only get worse at higher security levels” [9]. Though there exist several techniques [9] to convert pairing-based schemes from composite-order groups to prime-order groups, there is still a significant performance degradation due to the required size of the special vectors [21]. Therefore, it is desirable to construct an expressive CP-ABE scheme with partially hidden access structures using pairings in the prime-order groups.

In this paper, we focus on designing an expressive CP-ABE scheme in the prime-order groups which can hide attribute values from access structures. We compare our CP-ABE scheme with partially hidden access structures to others in the literature in Table 1. It is straightforward to see that our construction is comparable to the existing ones in that it allows unbounded attribute names, supports expressive access structures and is built in the prime-order groups.

### 1.1 Challenges and Our Contributions

In the real world, the attribute values always contain more sensitive information than the generic attribute names. For example, the attribute values “Cardiologist” and “NR005289” are more sensitive than the attribute names “Doctor” and “Patient”, respectively. Due to this observation, a notion called CP-ABE with partially hidden access structures [15, 19] was proposed which divides each attribute into an attribute name and an attribute value, and hides attribute values associated with an access structure included in a ciphertext. That is, instead of a full access structure, a partially hidden access structure (e.g., “(Patient: \* AND Hospital: \*) OR (Doctor: \* AND Hospital: \*)”) which consists of only attribute names without attribute values is attached to a ciphertext.

We build a CP-ABE scheme with partially hidden access structures from the large universe CP-ABE scheme proposed by Rouselakis and Waters [21], which is an unbounded CP-ABE scheme supporting expressive access policies in the prime-order groups. A naive approach to construct a CP-ABE scheme with partially hidden access structures is simply removing the attribute names from the access structure in the Rouselakis-Waters scheme. However, the resulting scheme suffers off-line dictionary attacks<sup>1</sup>. Therefore, the key challenge here is to modify the Rouselakis-Waters scheme [21] such that its access structure is partially hidden and secure against off-line dictionary attacks. Thanks to the “randomness splitting” technique [6], we build a CP-ABE scheme where the sensitive attribute values are hidden to a computationally bounded adversary by performing some sort of blinding through splitting each attribute value into two randomized complementary components. Thus, though the ciphertext and access structure still contain information about generic attribute names, attribute values are protected from off-line dictionary attacks.

However, since an attribute name in practice may correspond to a number of attribute values, a ciphertext with hidden attribute values raises another issue: given solely attribute names associated with an access structure in a ciphertext, how could a user know he/she is a privileged recipient or not? One solution to this

---

<sup>1</sup> We will show how an off-line dictionary attack works in Sect. 4.

problem is to also encrypt a publicly known message such as the unity element “1” in addition to the encryption of the real data, all under the same access structure [15, 24], but this almost doubles the size of the original ciphertext, which is undesirable to a cloud storage system who prefers to save storage space. To reduce the storage cost of cloud services, we simply make a commitment to the encrypted message, and thus a user can know whether he/she has access to the encrypted data by checking whether the decryption result is consistent with the given commitment of the underlying message.

In a nutshell, the differences between our construction of CP-ABE with partially hidden access structure and the Rouselakis-Waters CP-ABE scheme are threefold. Firstly, we perform a “linear splitting” technique [6] on various portions of a ciphertext to overcome the off-line dictionary attacks. Secondly, we re-randomize the key components upon each attribute to make the linear splitting methodology feasible for all attribute values appearing in the ciphertext. Thirdly, we make a commitment to the message to allow a user to check whether he/she is a privileged recipient of a ciphertext without knowing the attribute values ascribed to the ciphertext.

## 1.2 Related Work

Sahai and Waters [22] introduced a notion called attribute-based encryption (ABE), and then Goyal et al. [11] formulated key-policy ABE (KP-ABE) and CP-ABE as two complimentary forms of ABE. In a CP-ABE system, the private keys are associated with the sets of attributes and the ciphertexts are associated with the access policies, while the situation is reversed in a KP-ABE system. Nevertheless, we believe that KP-ABE is less flexible than CP-ABE because the access policy is determined once the user’s attribute-based private key is issued. Bethencourt, Sahai and Waters [3] proposed the first CP-ABE construction, but it was secure under the generic group model. Cheung and Newport [7] presented a CP-ABE scheme that was proved to be secure under the standard model, but it only supported the AND access structures. A CP-ABE system under more advanced access structures was proposed by Goyal et al. [10] based on the number theoretic assumption. Rouselakis and Waters [21] built a large universe CP-ABE system under the prime-order groups to overcome the limitation that the size of the attribute space is polynomially bounded. The cryptographic primitive of CP-ABE with partially hidden access structures was introduced by Nishide et al. [19], but their construction only admitted admissible access structures expressed in AND gates and is selectively secure. Following the work in [19], Li et al. [18] extended the construction with an additional property as user accountability. With the aim of improving efficiency in [18, 19], Zhang et al. [24] presented a methodology to reduce the computational overhead in the decryption, but their construction still did not support advanced access structures. Lai, Deng and Li [14] put forth a fully secure CP-ABE scheme with partially hidden access structures, but it only supports restricted access structures as in [18, 19]. Later, Lai, Deng and Li proposed [15] a fully secure CP-ABE scheme which can partially

hide access structures of any boolean formulas, but it was built from bilinear pairings in the composite-order groups.

### 1.3 Organization

The remainder of this paper is organized as follows. In Sect. 2, we briefly review the notions and definitions relevant to this paper. In Sect. 3, after depicting the framework for CP-ABE with partially hidden access structures, we present its security model. In Sect. 4, we give a concrete expressive and unbounded CP-ABE scheme with partially hidden access policies and analyze its security and performance. We conclude the paper in Sect. 5.

## 2 Preliminaries

In this section, we review some basic cryptographic notions and definitions that are to be used in this paper.

### 2.1 Bilinear Pairings and Complexity Assumptions

Let  $G$  be a group of prime order  $p$  that is generated from  $g$ . We define  $\hat{e} : G \times G \rightarrow G_1$  to be a bilinear map if it has the following properties [5]:

- Bilinear such that for all  $g \in G$ , and  $a, b \in Z_p$ , we have  $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$
- Non-degenerate such that  $\hat{e}(g, g) \neq 1$ .

We say that  $G$  is a bilinear group if the group operation in  $G$  is efficiently computable and there exists a group  $G_1$  and an efficiently computable bilinear map  $\hat{e} : G \times G \rightarrow G_1$  as above.

**Decisional  $(q-1)$  Assumption** [21]. The decisional  $(q-1)$  problem is that for any probabilistic polynomial-time algorithm, given  $\vec{y} =$

$$\begin{aligned} &g, g^\mu, \\ &g^{a^i}, g^{b_j}, g^{\mu \cdot b_j}, g^{a^i b_j}, g^{a^i / b_j^2} \quad \forall (i, j) \in [q, q], \\ &g^{a^i / b_j} \quad \forall (i, j) \in [2q, q] \text{ with } i \neq q+1, \\ &g^{a^i b_j / b_{j'}^2}, \quad \forall (i, j, j') \in [2q, q, q] \text{ with } j \neq j', \\ &g^{\mu a^i b_j / b_{j'}}, g^{\mu a^i b_j / b_{j'}^2} \quad \forall (i, j, j') \in [q, q, q] \text{ with } j \neq j', \end{aligned}$$

it is difficult to distinguish  $(\vec{y}, \hat{e}(g, g)^{a^{q+1}\mu})$  from  $(\vec{y}, Z)$ , where  $g \in G$ ,  $Z \in G_1$ ,  $a, \mu, b_1, \dots, b_q \in Z_p$  are chosen independently and uniformly at random.

**Decisional Linear Assumption** [4]. The decisional linear problem is that for any probabilistic polynomial-time algorithm, given  $g, g^{x_1}, g^{x_2}, g^{x_1 x_3}, g^{x_2 x_4}$ , it is difficult to distinguish  $(g, g^{x_1}, g^{x_2}, g^{x_1 x_3}, g^{x_2 x_4}, g^{x_3 + x_4})$  from  $(g, g^{x_1}, g^{x_2}, g^{x_1 x_3}, g^{x_2 x_4}, Z)$ , where  $g, Z \in G$ ,  $x_1, x_2, x_3, x_4 \in Z_p$  chosen independently and uniformly at random.

## 2.2 Access Structures and Linear Secret Sharing

We review the the notions of access structures and linear secret sharing schemes [17,23] as follows.

**Access Structures.** Let  $\{P_1, \dots, P_n\}$  be a set of parties. A collection  $\mathbb{A} \subseteq 2^{\{P_1, \dots, P_n\}}$  is monotone if  $\forall B, C : \text{if } B \in \mathbb{A} \text{ and } B \subseteq C, \text{ then } C \in \mathbb{A}$ . An (monotone) access structure is a (monotone) collection  $\mathbb{A}$  of non-empty subsets of  $\{P_1, \dots, P_n\}$ , i.e.,  $\mathbb{A} \subseteq 2^{\{P_1, \dots, P_n\}} \setminus \{\emptyset\}$ . The sets in  $\mathbb{A}$  are called the authorized sets, and the sets that are not in  $\mathbb{A}$  are called the unauthorized sets.

**Linear Secret Sharing Schemes (LSSSs).** Let  $P$  be a set of parties. Let  $\mathbb{M}$  be a matrix of size  $l \times n$ . Let  $\rho : \{1, \dots, l\} \rightarrow P$  be a function that maps a row to a party for labeling. A secret sharing scheme  $\Pi$  over a set of parties  $P$  is a linear secret-sharing scheme over  $Z_p$  if

1. the shares for each party form a vector over  $Z_p$ ;
2. there exists a matrix  $\mathbb{M}$  which has  $l$  rows and  $n$  columns called the share-generating matrix for  $\Pi$ . For  $x = 1, \dots, l$ , the  $x$ -th row of matrix  $\mathbb{M}$  is labeled by a party  $\rho(i)$ , where  $\rho : \{1, \dots, l\} \rightarrow P$  is a function that maps a row to a party for labeling. Considering that the column vector  $v = (\mu, r_2, \dots, r_n)$ , where  $\mu \in Z_p$  is the secret to be shared and  $r_2, \dots, r_n \in Z_p$  are randomly chosen, then  $\mathbb{M}v$  is the vector of  $l$  shares of the secret  $\mu$  according to  $\Pi$ . The share  $(\mathbb{M}v)_i$  belongs to party  $\rho(i)$ .

It has been noted in [17] that every LSSS also enjoys the linear reconstruction property. Suppose that  $\Pi$  is an LSSS for access structure  $\mathbb{A}$ . Let  $\mathbf{A}$  be an authorized set, and define  $I \subseteq \{1, \dots, l\}$  as  $I = \{i | \rho(i) \in \mathbf{A}\}$ . Then the vector  $(1, 0, \dots, 0)$  is in the span of rows of matrix  $\mathbb{M}$  indexed by  $I$ , and there exist constants  $\{w_i \in Z_p\}_{i \in I}$  such that, for any valid shares  $\{v_i\}$  of a secret  $\mu$  according to  $\Pi$ , we have  $\sum_{i \in I} w_i v_i = \mu$ . These constants  $\{w_i\}$  can be found in polynomial time with respect to the size of the share-generating matrix  $\mathbb{M}$  [2].

On the other hand, for an unauthorized set  $\mathbf{A}'$ , no such constants  $\{w_i\}$  exist. Moreover, in this case it is also true that if  $I' = \{i | \rho(i) \in \mathbf{A}'\}$ , there exists a vector  $\vec{w}$  such that its first component  $w_1$  is any non zero element in  $Z_p$  and  $\langle \mathbb{M}_i, \vec{w} \rangle = 0$  for all  $i \in I'$ , where  $\mathbb{M}_i$  is the  $i$ -th row of  $\mathbb{M}$  [21].

**Boolean Formulas** [17]. Access policies can also be described in terms of monotonic boolean formulas. LSSS access structures are more general, and can be derived from representations as boolean formulas. There are standard techniques to convert any monotonic boolean formula into a corresponding LSSS matrix. The boolean formula can be represented as an access tree, where the interior nodes are AND and OR gates, and the leaf nodes correspond to attributes. The number of rows in the corresponding LSSS matrix will be the same as the number of leaf nodes in the access tree.

### 3 System Architecture and Security Model

In this section, we describe the framework and security model of ciphertext-policy attribute-based encryption with partially hidden access structures.

#### 3.1 Framework

A CP-ABE scheme with partially hidden access structures consists of four algorithms: setup algorithm  $\text{Setup}$ , attribute-based private key generation algorithm  $\text{KeyGen}$ , encryption algorithm  $\text{Encrypt}$  and decryption algorithm  $\text{Decrypt}$ .

- $\text{Setup}(1^\lambda) \rightarrow (\text{pars}, \text{msk})$ . Taking the security parameter  $\lambda$  as the input, this algorithm outputs the public parameter  $\text{pars}$  and the master private key  $\text{msk}$  for the system. This algorithm is run by the AA.
- $\text{KeyGen}(\text{pars}, \text{msk}, \mathbf{A}) \rightarrow K_{\mathbf{A}}$ . Taking the public parameter  $\text{pars}$ , the master private key  $\text{msk}$  and an attribute set  $\mathbf{A}$  as the input, this algorithm outputs an attribute-based private key  $K_{\mathbf{A}}$  over the attribute set  $\mathbf{A}$ . This algorithm is run by the AA.
- $\text{Encrypt}(\text{pars}, M, (\mathbb{M}, \rho, \{A_{\rho(i)}\})) \rightarrow \text{CT}$ . Taking the public parameter  $\text{pars}$ , a message  $M$  and an access structure  $(\mathbb{M}, \rho, \{A_{\rho(i)}\})$  where the function  $\rho$  associates the rows of  $\mathbb{M}$  to generic attribute names, and  $\{A_{\rho(i)}\}$  are the corresponding attribute values as the input. Let  $\mathbb{M}$  be an  $l \times n$  matrix as the input, this algorithm outputs a ciphertext  $\text{CT}$ . This algorithm is run by the data owner.
- $\text{Decrypt}(\text{pars}, \text{CT}, \mathbf{A}, K_{\mathbf{A}}) \rightarrow M/\perp$ . Taking the public parameter  $\text{pars}$ , a ciphertext  $\text{CT}$  and an attribute-based private key  $K_{\mathbf{A}}$  associated to an attribute set  $\mathbf{A}$  as the input, this algorithm outputs either the message  $M$  when the private key  $K_{\mathbf{A}}$  satisfies the access structure, or a symbol  $\perp$  otherwise. This algorithm is run by the user.

We require that a CP-ABE scheme with partially hidden access structures is correct, meaning that for all messages  $M$ , all attribute sets  $\mathbf{A}$  and access structures  $(\mathbb{M}, \rho, \{A_{\rho(i)}\})$  with authorized  $\mathbf{A}$  satisfying  $(\mathbb{M}, \rho, \{A_{\rho(i)}\})$ , if  $(\text{pars}, \text{msk}) \leftarrow \text{Setup}(1^\lambda)$ ,  $K_{\mathbf{A}} \leftarrow \text{KeyGen}(\text{pars}, \text{msk}, \mathbf{A})$ ,  $\text{CT} \leftarrow \text{Encrypt}(\text{pars}, M, (\mathbb{M}, \rho, \{A_{\rho(i)}\}))$ , then  $\text{Decrypt}(\text{pars}, \text{CT}, \mathbf{A}, K_{\mathbf{A}}) = M$ .

#### 3.2 Security Definitions

A CP-ABE scheme with partially hidden access structures should ensure confidentiality and anonymity. Below we elaborately describe the security definitions for these two requirements one by one.

**Confidentiality.** Assuming that the adversary makes the key generation queries adaptively, we define the security model for confidentiality by the following game between a challenger algorithm  $\mathcal{C}$  and an adversary algorithm  $\mathcal{A}$ , based on the security model of indistinguishability under chosen-plaintext attacks (IND-CPA) for CP-ABE [23].

- Setup. Algorithm  $\mathcal{C}$  runs the setup algorithm, and gives the public parameter  $pars$  to algorithm  $\mathcal{A}$  and keeps the master private key  $msk$ .
- Phase 1. Algorithm  $\mathcal{A}$  makes the key generation queries to algorithm  $\mathcal{C}$ . Algorithm  $\mathcal{C}$  sends an attribute set  $\mathbf{A}_i$  to algorithm  $\mathcal{C}$ . Algorithm  $\mathcal{C}$  responds by returning the corresponding key  $K_{\mathbf{A}_i}$  to algorithm  $\mathcal{A}$ .
- Challenge. Algorithm  $\mathcal{A}$  chooses two messages  $M_0^*$  and  $M_1^*$  of the same size, and an access structure  $(\mathbb{M}^*, \rho^*, \{A_{\rho(i)}^*\})$  with the constraint that the key generation queries  $\{K_{\mathbf{A}_i}\}$  in Phase 1 do not satisfy the access structure  $(\mathbb{M}^*, \rho^*, \{A_{\rho(i)}^*\})$ . The challenger chooses a random bit  $\beta \in \{0, 1\}$ , and sends algorithm  $\mathcal{A}$  a challenge ciphertext  $CT^*$  which is an encryption of  $M_\beta^*$  under the access structure  $(\mathbb{M}^*, \rho^*, \{A_{\rho(i)}^*\})$ .
- Phase 2. Algorithm  $\mathcal{A}$  continues issuing the key generation queries on attribute sets  $\mathbf{A}_i$  with the constraint that they do not satisfy the access structure in the challenge phase. Algorithm  $\mathcal{C}$  responds as in Phase 1.
- Guess. Algorithm  $\mathcal{A}$  makes a guess  $\beta'$  for  $\beta$ , and it wins the game if  $\beta' = \beta$ .

**Anonymity.** Anonymity prevents an adversary from distinguishing a ciphertext under one access matrix associated with one attribute set from a ciphertext under the same access matrix associated with another attribute set. In the anonymity game, the adversary is given the public parameter, as well as the access to the key generation oracle, and its goal is to guess which of two attribute sets satisfying the same access matrix generates the ciphertext in the challenge phase, without being given either of the private keys associated with the two attribute sets. Below we define the the game of anonymity under chosen-plaintext attacks (ANON-CPA) between a challenger algorithm  $\mathcal{C}$  and an adversary algorithm  $\mathcal{A}$ .

- Setup. Algorithm  $\mathcal{C}$  runs the setup algorithm, and gives the public parameter  $pars$  to algorithm  $\mathcal{A}$  and keeps the master private key  $msk$ .
- Phase 1. Algorithm  $\mathcal{A}$  makes the key generation query to algorithm  $\mathcal{C}$ . Algorithm  $\mathcal{C}$  sends an attribute set  $\mathbf{A}_i$  to algorithm  $\mathcal{C}$ . Algorithm  $\mathcal{C}$  responds by returning the corresponding key  $K_{\mathbf{A}_i}$  to algorithm  $\mathcal{A}$ .
- Challenge. Algorithm  $\mathcal{A}$  chooses a message  $M^*$  and an access matrix  $(\mathbb{M}^*, \rho^*)$  which can be satisfied by attribute sets  $\{A_{\rho(i)}^*\}_0$  and  $\{A_{\rho(i)}^*\}_1$  with the constraint that there are no key generation queries  $\{K_{\mathbf{A}_i}\}$  in Phase 1 that can satisfy  $(\mathbb{M}^*, \rho^*, \{A_{\rho(i)}^*\}_0)$  and  $(\mathbb{M}^*, \rho^*, \{A_{\rho(i)}^*\}_1)$ . The challenger chooses a random bit  $\beta \in \{0, 1\}$ , and sends algorithm  $\mathcal{A}$  a challenge ciphertext  $CT^*$  which is an encryption of  $M^*$  under the access structure  $(\mathbb{M}^*, \rho^*, \{A_{\rho(i)}^*\}_\beta)$ .
- Phase 2. Algorithm  $\mathcal{A}$  continues issuing the key generation queries to algorithm  $\mathcal{C}$ . Algorithm  $\mathcal{C}$  responds as in Phase 1 with the constraint that the attributes of the key generation queries satisfying  $(\mathbb{M}^*, \rho^*, \{A_{\rho(i)}^*\}_0)$  and  $(\mathbb{M}^*, \rho^*, \{A_{\rho(i)}^*\}_1)$  are disallowed. Algorithm  $\mathcal{C}$  responds as in Phase 1.
- Guess. Algorithm  $\mathcal{A}$  makes a guess  $\beta'$  for  $\beta$ , and it wins the game if  $\beta' = \beta$ .

Algorithm  $\mathcal{A}$ 's advantage in the above two games are defined as  $\Pr[\beta = \beta'] - 1/2$ . We say that a CP-ABE scheme with partially hidden access structures is indistinguishable (or anonymous) under the chosen-plaintext attacks if



all probabilistic polynomial time (PPT) adversaries have at most a negligible advantage in the security parameter  $\lambda$ . In addition, a CP-ABE scheme with partially hidden access structures is said to be selectively indistinguishable (or anonymous) if an Init stage is added before the Setup phase where algorithm  $\mathcal{A}$  commits to the challenge access structure  $(\mathbb{M}, \rho, \{A_{\rho(i)}\})$ .

## 4 Ciphertext-Policy Attribute-Based Encryption Scheme with Partially Hidden Access Structures

In this section, we give a concrete construction of a CP-ABE scheme with partially hidden access structures, and analyze its security and performance.

### 4.1 Attribute Value Guessing Attack

Below we briefly review the encryption algorithm of the CP-ABE scheme in [21], and show that there is an attribute value guessing attack to such a construction.

**Encrypt.** This algorithm takes the public parameter  $pars$ , a message  $M$  and an LSSS access structure  $(\mathbb{M}, \rho)$  where the function  $\rho$  associates the rows of  $\mathbb{M}$  to attributes as the input. Let  $\mathbb{M}$  be an  $l \times n$  matrix. It randomly chooses a vector  $\vec{v} = (\mu, y_2, \dots, y_n) \in Z_p^n$ . These values will be used to share the encryption exponent  $\mu$ . For  $i = 1$  to  $l$ , it calculates  $v_i = \vec{v} \cdot \mathbb{M}_i$ , where  $\mathbb{M}_i$  is the vector corresponding to the  $i$ -th row of  $\mathbb{M}$ . In addition, it randomly chooses  $\beta, z_1, \dots, z_l \in Z_p$ , and outputs a ciphertext  $CT = (C, D, \{(C_i, D_i, E_i)\}_{i \in [1, l]})$ .

$$C = \hat{e}(g, g)^{\alpha\mu}, D = g^\mu, C_i = w^{v_i} v^{z_i}, D_i = g^{z_i}, E_i = (u^{\rho(i)} h)^{-z_i},$$

where  $g, u, h, v, w, \hat{e}(g, g)^\alpha$  belong to the public parameter  $pars$ .

**Attack.** Given a ciphertext  $CT = (C, D, \{(C_i, D_i, E_i)\}_{i \in [1, l]})$ , an adversary can easily determine whether an attribute value  $A_i$  used in the ciphertext by checking whether  $\hat{e}(E_i, g) = \hat{e}(u^{A_i} h, D_i^{-1})$  holds. Clearly, this scheme cannot achieve anonymity.

### 4.2 Construction

On the basis of the large universe CP-ABE scheme proposed in [21], we present a CP-ABE scheme which can partially hide the access structures in the prime-order groups. Let  $G$  be a bilinear group of a prime order  $p$  with a generator  $g$ . Denote  $\hat{e} : G \times G \rightarrow G_1$  by the bilinear map.

- **Setup.** This algorithm takes the security parameter  $\lambda$  as the input. It randomly chooses a group  $G$  of prime order  $p$  with a generator  $g$ . Also, it randomly chooses  $u, h, v, w \in G, d_1, d_2, d_3, d_4, \alpha \in Z_p$ , and computes  $g_1 = g^{d_1}, g_2 = g^{d_2}, g_3 = g^{d_3}, g_4 = g^{d_4}$ . The public parameter is  $pars = (H, g, u, h, w, v, g_1, g_2, g_3, g_4, \hat{e}(g, g)^\alpha)$  where  $H$  is a collision resistant hash function that maps an element in  $G_1$  to an element in  $\{0, 1\}^t$  with  $t$  being the security parameter such that the concatenate elements in  $Z_p$  are represented in  $t$  bits, and the master private key is  $msk = (d_1, d_2, d_3, d_4, g^\alpha)$ .

- KeyGen. This algorithm takes the public parameter  $pars$ , the master private key  $msk$  and an attribute set  $\mathbf{A}^2$  as the input. Let  $k$  be the size of  $\mathbf{A}$ , and  $A_1, \dots, A_k \in Z_p$  be the attribute values of  $\mathbf{A}$ . It randomly chooses  $r, r', r_1, \dots, r_k, r'_1, \dots, r'_k \in Z_p$ , and outputs the attribute-based private key  $K_{\mathbf{A}} = (K_1, K_2, \{K_{i,1}, K_{i,2}, K_{i,3}, K_{i,4}, K_{i,5}\}_{i \in [1,k]})$  over a set of attributes  $\mathbf{A}$  as

$$\begin{aligned} K_1 &= g^{\alpha} w^{d_1 d_2 r + d_3 d_4 r'}, & K_2 &= g^{r d_1 d_2 + r' d_3 d_4}, \\ K_{i,1} &= ((u^{A_i} h)^{r_i} v^{-r})^{d_2}, & K_{i,2} &= ((u^{A_i} h)^{r_i} v^{-r})^{d_1}, & K_{i,3} &= g^{d_1 d_2 r_i + d_3 d_4 r'_i}, \\ K_{i,4} &= ((u^{A_i} h)^{r'_i} v^{-r'})^{d_4}, & K_{i,5} &= ((u^{A_i} h)^{r'_i} v^{-r'})^{d_3}. \end{aligned}$$

- Encrypt. This algorithm takes the public parameter  $pars$ , a message  $M \in Z_p$  and an LSSS access structure  $(\mathbb{M}, \rho, \{A_{\rho(i)}\})^3$  as the input. It randomly chooses a vector  $\vec{v} = (\mu, y_2, \dots, y_n) \in Z_p^n$ . These values will be used to share the encryption exponent  $\mu$ . For  $i = 1$  to  $l$ , it calculates  $v_i = \vec{v} \cdot \mathbb{M}_i$ , where  $\mathbb{M}_i$  is the vector corresponding to the  $i$ -th row of  $\mathbb{M}$ . Then, it randomly chooses  $\gamma, s_{i,1}, \dots, s_{i,l}, s_{1,2}, \dots, s_{l,2}, z_1, \dots, z_l \in Z_p$ , and outputs a ciphertext  $CT = ((\mathbb{M}, \rho), C, D, E, \{(C_i, D_{i,1}, D_{i,2}, E_{i,1}, E_{i,2}, F_i)\}_{i \in [1,l]})$ , where

$$\begin{aligned} C &= (M || \gamma) \oplus H(\hat{e}(g, g)^{\alpha \mu}), & D &= g^{\mu}, & E &= g^M h^{\gamma}, \\ C_i &= w^{v_i} v^{z_i}, & D_{i,1} &= g_1^{z_i - s_{i,1}}, & D_{i,2} &= g_3^{z_i - s_{i,2}}, \\ E_{i,1} &= g_2^{s_{i,1}}, & E_{i,2} &= g_4^{s_{i,2}}, & F_i &= (u^{A_{\rho(i)}} h)^{-z_i}. \end{aligned}$$

- Decrypt. This algorithm takes the public parameter  $pars$ , a ciphertext  $((\mathbb{M}, \rho), C, D, E, \{(C_i, D_{i,1}, D_{i,2}, E_{i,1}, E_{i,2}, F_i)\}_{i \in [1,l]})$  and a private key  $K_{\mathbf{A}}$  for an attribute set  $\mathbf{A}$  as the input. It calculates  $I_{\mathbb{M}, \rho}$  from  $(\mathbb{M}, \rho)$ , which is a set of minimum subsets of attributes satisfying  $(\mathbb{M}, \rho)$ . Denote by  $\{w_i \in Z_p\}_{i \in \mathcal{I}}$  a set of constants such that if  $\{v_i\}$  are valid shares of any secret  $\mu$  according to  $\mathbb{M}$ , then  $\sum_{i \in \mathcal{I}} w_i v_i = \mu$ . For an  $\mathcal{I} \in I_{\mathbb{M}, \rho}$ , it computes

$$\begin{aligned} & \frac{\hat{e}(D, K_1)}{\prod_{i \in \mathcal{I}} (\hat{e}(C_i, K_2) \hat{e}(D_{i,1}, K_{i,1}) \hat{e}(E_{i,1}, K_{i,2}) \hat{e}(F_i, K_{i,3}) \hat{e}(D_{i,2}, K_{i,4}) \hat{e}(E_{i,2}, K_{i,5}))^{w_i}} \\ &= \frac{\hat{e}(g, g)^{\alpha \mu} \hat{e}(g^{\mu}, w)^{r_1 d_1 d_2} \hat{e}(g^{\mu}, w)^{r_2 d_3 d_4}}{\prod_{i \in \mathcal{I}} (\hat{e}(g, w^{v_i})^{d_1 d_2 r_1 + d_3 d_4 r_2})^{w_i}} = \hat{e}(g, g)^{\alpha \mu}, & \frac{C}{H(\hat{e}(g, g)^{\alpha \mu})} &= M || \gamma. \end{aligned}$$

If  $g^M h^{\gamma} = E$ , it outputs  $M$ . Otherwise, it outputs  $\perp$ .

**Remarks.** In the above construction, the term  $E$ , computed using a commitment scheme [20], is added to the ciphertext such that a user can easily ascertain whether he/she is a privileged recipient by checking the decryption result via the given  $E$ . Note that according to the binding property of the commitment scheme [8], each  $E$  can only be obtained from a unique pair of  $M$  and  $\gamma$ , which

<sup>2</sup> Note that each attribute is denoted as  $N_i = A_i$ , where  $N_i$  is the generic name of an attribute and  $A_i$  is the corresponding attribute value.

<sup>3</sup> For the details about how to convert a boolean formula into an equivalent LSSS matrix, please refer to [17].

guarantees the correctness of decryption, in spite of the fact that the user has no idea whether his/her attribute set satisfies the access structure ascribed the ciphertext before performing decryption.

### 4.3 Security Proof

**Theorem 1.** *Assuming that the  $(q - 1)$  assumption holds in  $G$ , and the decisional linear assumption holds in  $G$ , then the above system is selectively indistinguishable and anonymous.*

*Proof.* At a high level, the proof is reduced via a sequence of games by concluding that these games are computationally indistinguishable from each other. For succinct description, we remove the access structure related elements from the ciphertext. Denote  $(C^*, D^*, E^*, \{(C_i^*, D_{i,1}^*, D_{i,2}^*, E_{i,1}^*, E_{i,2}^*, F_i^*)\}_{i \in [1, l]})$  by the challenge ciphertext given to the adversary during an attack in the real world. Let  $Z$  be a random element of  $G_1$ , and  $\{Z_{i,1}\}, \{Z'_{i,1}\}$  be sets of random elements of  $G$ . We define a sequence of games  $\text{Game}_0, \text{Game}_1, \dots, \text{Game}_l, \text{Game}_{l+1}, \dots, \text{Game}_{2l+1}$  that differ on which challenge ciphertext is given by the challenger to the adversary, where  $\text{Game}_0$  is the original game,  $\text{Game}_1$  changes the term  $C^*$  to  $Z$ , and  $\text{Game}_2$  to  $\text{Game}_{l+1}$  change the  $D_{i,1}^*$  term to  $Z_{i,1}$  one by one for  $i \in [1, l]$ , and  $\text{Game}_{l+2}$  to  $\text{Game}_{2l+1}$  change the  $E_{i,1}^*$  term to  $Z'_{i,1}$  one by one for  $i \in [1, l]$ .

- $\text{Game}_0$ : The challenge ciphertext is  $\text{CT}_0^* = (C^*, D^*, E^*, \{(C_i^*, D_{i,1}^*, D_{i,2}^*, E_{i,1}^*, E_{i,2}^*, F_i^*)\}_{i \in [1, l]})$ .
- $\text{Game}_1$ : The challenge ciphertext is  $\text{CT}_1^* = (Z, D^*, E^*, \{(C_i^*, D_{i,1}^*, D_{i,2}^*, E_{i,1}^*, E_{i,2}^*, F_i^*)\}_{i \in [1, l]})$ .
- $\text{Game}_2$ : The challenge ciphertext is  $\text{CT}_2^* = (Z, D^*, E^*, (C_1, Z_{1,1}, D_{1,2}^*, E_{1,1}^*, E_{1,2}^*, F_1^*), \{(C_i^*, D_{i,1}^*, D_{i,2}^*, E_{i,1}^*, E_{i,2}^*, F_i^*)\}_{i \in [2, l]})$ .
- $\dots$
- $\text{Game}_{l+1}$ : The challenge ciphertext is  $\text{CT}_{l+1}^* = (Z, D^*, E^*, \{(C_i, Z_{i,1}, D_{i,2}^*, E_{i,1}^*, E_{i,2}^*, F_i^*)\}_{i \in [1, l]})$ .
- $\text{Game}_{l+2}$ : The challenge ciphertext is  $\text{CT}_{l+2}^* = (Z, D^*, E^*, (C_1^*, Z_{1,1}, Z'_{1,1}, E_{1,1}^*, E_{1,2}^*, F_1^*), \{(C_i^*, Z_{i,1}, D_{i,2}^*, E_{i,1}^*, E_{i,2}^*, F_i^*)\}_{i \in [2, l]})$ .
- $\dots$
- $\text{Game}_{2l+1}$ : The challenge ciphertext is  $\text{CT}_{2l+1}^* = (Z, D^*, E^*, \{(C_i^*, Z_{i,1}, Z'_{i,1}, E_{i,1}^*, E_{i,2}^*, F_i^*)\}_{i \in [1, l]})$ .

To complete the proof, we will show that the games  $\text{Game}_0, \text{Game}_1, \dots, \text{Game}_{2l+1}$  are computationally indistinguishable.

**Lemma 1.** *Assuming that the  $(q - 1)$  assumption holds in  $G$ , then there is no adversary that distinguishes between the games  $\text{Game}_0$  and  $\text{Game}_1$ .*

*Proof.* Assume that there exists an adversary algorithm  $\mathcal{A}$  that can distinguish  $\text{Game}_0$  from  $\text{Game}_1$ . Then we can build a challenger algorithm  $\mathcal{C}$  that solves the  $(q - 1)$  problem.

- Init. Algorithm  $\mathcal{A}$  gives algorithm  $\mathcal{C}$  a challenge access structure  $(\mathbb{M}^*, \rho^*, \{\rho(i)^*\})$ <sup>4</sup>, where  $\mathbb{M}^*$  is an  $l \times n$  matrix.
- Setup. Algorithm  $\mathcal{C}$  randomly chooses  $d_1, d_2, d_3, d_4 \in Z_p$ , and computes  $g_1 = g^{d_1}, g_2 = g^{d_2}, g_3 = g^{d_3}, g_4 = g^{d_4}$ . Then, it randomly chooses a hash function  $H: G_1 \rightarrow \{0, 1\}^t$ ,  $\tilde{\alpha}, \tilde{u}, \tilde{v}, \tilde{h} \in Z_p$ . In addition, it implicitly sets  $\alpha = a^{q+1} + \tilde{\alpha}$ , and outputs the rest of the public parameter as  $g = g, w = g^a$ ,

$$v = g^{\tilde{v}} \cdot \prod_{(j,j') \in [l,n]} (g^{a^{j'}/b_j})^{\mathbb{M}_{j,j'}^*}, \quad u = g^{\tilde{u}} \cdot \prod_{(j,j') \in [l,n]} (g^{a^{j'}/b_j^2})^{\mathbb{M}_{j,j'}^*},$$

$$h = g^{\tilde{h}} \cdot \prod_{(j,j') \in [l,n]} (g^{a^{j'}/b_j^2})^{-\rho^*(j)\mathbb{M}_{j,j'}^*}, \quad \hat{e}(g, g)^\alpha = \hat{e}(g^a, g^{a^q}) \cdot \hat{e}(g, g)^{\tilde{\alpha}}.$$

- Phase 1 and Phase 2. In both phases, algorithm  $\mathcal{C}$  has to output the private keys for attribute sets  $\mathbf{A} = \{A_1, \dots, A_{|\mathbf{A}|}\}$  issued by algorithm  $\mathcal{A}$ .

Since  $\mathbf{A}$  does not satisfy  $(\mathbb{M}^*, \rho^*, \{\rho(i)^*\})$ , there exists a vector  $\vec{w} = (w_1, \dots, w_n)^\perp \in Z_p^n$  such that  $w_1 = -1$ ,  $(\mathbb{M}_i^*, \vec{w}) = 0$  for all  $i \in I = \{i | i \in [l] \wedge \rho(i)^* \in \mathbf{A}\}$ . Algorithm  $\mathcal{B}$  computes  $\vec{w}$  using linear algebra. In addition, it randomly chooses  $\tilde{r}, \tilde{r}' \in Z_p$ , implicitly sets

$$r = \tilde{r} + w_1 a^q + w_2 a^{q-1} + \dots + w_n a^{q+1-n} = \tilde{r} + \sum_{i \in [n]} w_i a^{q+1-i},$$

$$r' = \tilde{r}' + w_1 a^q + w_2 a^{q-1} + \dots + w_n a^{q+1-n} = \tilde{r}' + \sum_{i \in [n]} w_i a^{q+1-i},$$

and computes

$$K_1 = g^\alpha w^{d_1 d_2 r + d_3 d_4 r'}$$

$$= (g^{a^{q+1}} g^{\tilde{\alpha}}) (g^{a\tilde{r}} \prod_{i \in [n]} g^{w_i a^{q+2-i}})^{d_1 d_2} (g^{a\tilde{r}'} \prod_{i \in [n]} g^{w_i a^{q+2-i}})^{d_3 d_4},$$

$$K_2 = g^{d_1 d_2 r + d_3 d_4 r'}$$

$$= (g^{\tilde{r}} \prod_{i \in [n]} (g^{a^{q+1-i}})^{w_i})^{d_1 d_2} (g^{\tilde{r}'} \prod_{i \in [n]} (g^{a^{q+1-i}})^{w_i})^{d_3 d_4}.$$

Then it computes

$$v^{-r} = v^{-\tilde{r}} \cdot \prod_{i \in [n]} (g^{a^{q+1-i}})^{-\tilde{v} w_i}$$

$$\cdot \prod_{\substack{(i,j,j') \in [n,l,n] \\ i \neq j'}} (g^{a^{q+1+j'-i}/b_j})^{-w_i \mathbb{M}_{j,j'}^*} \prod_{\substack{j \in [l] \\ \rho(j) \notin \mathbf{A}}} g^{(\vec{w} \cdot \mathbb{M}_j^*) a^{q+1}/b_j},$$

$$v^{-r'} = v^{-\tilde{r}'} \cdot \prod_{i \in [n]} (g^{a^{q+1-i}})^{-\tilde{v} w_i}$$

$$\cdot \prod_{\substack{(i,j,j') \in [n,l,n] \\ i \neq j'}} (g^{a^{q+1+j'-i}/b_j})^{-w_i \mathbb{M}_{j,j'}^*} \prod_{\substack{j \in [l] \\ \rho(j) \notin \mathbf{A}}} g^{(\vec{w} \cdot \mathbb{M}_j^*) a^{q+1}/b_j},$$

<sup>4</sup> For notation simplicity, we use  $\{\rho(i)^*\}$  to replace  $\{A_{\rho(i)}^*\}$  in the rest of the proof.

where the last parts cannot be directly calculated, so it must be canceled by the  $(u^{A_i}h)^{r_i}$ ,  $(u^{A_i}h)^{r'_i}$  parts.

Therefore, for all  $i \in [[\mathbf{A}]]$ , algorithm  $\mathcal{C}$  randomly chooses  $\tilde{r}_i \in Z_p$ , and implicitly sets

$$r_i = \tilde{r}_i + (\tilde{r} \cdot \sum_{\substack{i' \in [l] \\ \rho^*(i') \notin \mathbf{A}}} \frac{b_j}{A_i - \rho^*(i')} + \sum_{\substack{j, i' \in [n, l] \\ \rho^*(i') \notin \mathbf{A}}} \frac{w_j b_{i'} a^{q+1-j}}{A_i - \rho^*(i')}),$$

$$r'_i = \tilde{r}'_i + (\tilde{r}' \cdot \sum_{\substack{i' \in [l] \\ \rho^*(i') \notin \mathbf{A}}} \frac{b_j}{A_i - \rho^*(i')} + \sum_{\substack{j, i' \in [n, l] \\ \rho^*(i') \notin \mathbf{A}}} \frac{w_j b_{i'} a^{q+1-j}}{A_i - \rho^*(i')}).$$

and computes

$$g^{r_i} = g^{\tilde{r}_i} \cdot \prod_{\substack{i' \in [l] \\ \rho^*(i') \notin \mathbf{A}}} (g^{b_{i'}})^{\frac{\tilde{r}}{A_i - \rho^*(i')}} \cdot \prod_{\substack{(k', i') \in [n, l] \\ \rho^*(i') \notin \mathbf{A}}} (g^{b_{i'} a^{q+1-k'}})^{\frac{w_{k'}}{A_i - \rho^*(i')}},$$

$$(u^{A_i}h)^{r_i} = (u^{A_i}h)^{\tilde{r}_i} \cdot \left(\frac{g^{r_i}}{g^{\tilde{r}_i}}\right)^{\tilde{u}A_i + \tilde{h}} \cdot \prod_{\substack{(i', j, j') \in [l, l, n] \\ \rho^*(i') \notin \mathbf{A}}} (g^{b_{i'} a^{j'} / b_j^2})^{\frac{\tilde{r}(A_i - \rho^*(j)) \mathbb{M}_{j, j'}^*}{A_i - \rho^*(i')}} \\ \cdot \prod_{\substack{(k', i', j, j') \in [n, l, l, n] \\ \rho^*(i') \notin \mathbf{A}, (j \neq i', k' \neq j')}} \left(g^{\frac{b_{i'} a^{q+1+j'} - k'}{b_j^2}}\right)^{\frac{A_i - \rho^*(j) w_{k'} \mathbb{M}_{j, j'}^*}{A_i - \rho^*(i')}} \\ \cdot \prod_{\substack{j \in [l] \\ \rho^*(j) \notin \mathbf{A}}} g^{\frac{(\vec{w} \cdot \mathbb{M}_j^*) a^{q+1}}{b_j}},$$

$$(u^{A_i}h)^{r'_i} = (u^{A_i}h)^{\tilde{r}'_i} \cdot \left(\frac{g^{r'_i}}{g^{\tilde{r}'_i}}\right)^{\tilde{u}A_i + \tilde{h}} \cdot \prod_{\substack{(i', j, j') \in [l, l, n] \\ \rho^*(i') \notin \mathbf{A}}} (g^{b_{i'} a^{j'} / b_j^2})^{\frac{\tilde{r}(A_i - \rho^*(j)) \mathbb{M}_{j, j'}^*}{A_i - \rho^*(i')}} \\ \cdot \prod_{\substack{(k', i', j, j') \in [n, l, l, n] \\ \rho^*(i') \notin \mathbf{A}, (j \neq i', k' \neq j')}} \left(g^{\frac{b_{i'} a^{q+1+j'} - k'}{b_j^2}}\right)^{\frac{A_i - \rho^*(j) w_{k'} \mathbb{M}_{j, j'}^*}{A_i - \rho^*(i')}} \\ \cdot \prod_{\substack{j \in [l] \\ \rho^*(j) \notin \mathbf{A}}} g^{\frac{(\vec{w} \cdot \mathbb{M}_j^*) a^{q+1}}{b_j}}.$$

Therefore, algorithm  $\mathcal{C}$  can output the private key  $K_{\mathbf{A}} = (K_1, K_2, \{K_{i,1}, K_{i,2}, K_{i,3}, K_{i,4}, K_{i,5}\}_{i \in [1, k]})$  for an attribute set  $\mathbf{A}$  as required.

- Challenge. Algorithm  $\mathcal{A}$  sends algorithm  $\mathcal{C}$  a message  $M^*$ . Algorithm  $\mathcal{C}$  randomly chooses  $\gamma \in Z_p$ , computes

$$C^* = (M^* || \gamma) \oplus H(Z \cdot \hat{e}(g, g^s)^{\tilde{\alpha}}), \quad D^* = g^s, \quad E^* = g^{M^*} h^\gamma.$$

Then it implicitly sets  $\vec{v} = (s, sa + \tilde{y}_2, sa^2 + \tilde{y}_3, \dots, sa^{n-1} + \tilde{y}_n)$ , where  $\tilde{y}_2, \dots, \tilde{y}_n \in Z_p$ , and

$$v_i = \sum_{j \in [n]} \mathbb{M}_{i,j}^* sa^{j-1} + \sum_{j=2}^n \mathbb{M}_{i,j}^* \tilde{y}_j = \sum_{j \in [n]} \mathbb{M}_{i,j}^* sa^{j-1} + \tilde{v}_i$$

for each row  $i \in [l]$ .

Additionally, it implicitly sets  $z_i = -sb_i$ , and computes

$$\begin{aligned} C_i^* &= w^{v_i} v^{z_i} = w^{\tilde{v}_i} \cdot \prod_{j \in [n]} g^{\mathbb{M}_{i,j}^* sa^j} \cdot (g^{sb_i})^{-\tilde{v}} \cdot \prod_{(i',j') \in [l,n]} g^{\frac{-\mathbb{M}_{i',j'}^* a^{j'}}{b_{i'}^{j'}} sb_i} \\ &= w^{\tilde{v}_i} \cdot (g^{sb_i})^{\tilde{v}} \cdot \prod_{\substack{(i',j') \in [l,n] \\ i' \neq i}} (g^{sa^{j'} b_i / b_{i'}})^{-\mathbb{M}_{i',j'}^*}, \\ F_i^* &= (u^{\rho^*(i)} h)^{z_i} = (g^{sb_i})^{-(\bar{u}\rho^*(i) + \tilde{h})} \cdot \left( \prod_{(i',j') \in [l,n]} g^{\frac{(\rho^*(i) - \rho^*(i')) \mathbb{M}_{i',j'}^* a^{j'}}{b_{i'}^{j'}}} \right)^{-sb_i} \\ &= (g^{sb_i})^{-(\bar{u}\rho^*(i) + \tilde{h})} \cdot \prod_{\substack{(i',j') \in [l,n] \\ i' \neq i}} (g^{\frac{sa^{j'} b_i}{b_{i'}^{j'}}})^{-(\rho^*(i) - \rho^*(i')) \mathbb{M}_{i',j'}^*}, \end{aligned}$$

$$\begin{aligned} D_{i,1}^* &= g_1^{z_i - s_{i,1}} = (g^{-sb_i})^{d_1} \cdot g^{-d_1 s_{i,1}}, & E_{i,1}^* &= g_2^{s_{i,1}} = g^{d_2 s_{i,1}}, \\ D_{i,2}^* &= g_3^{z_i - s_{i,2}} = (g^{-sb_i})^{d_3} \cdot g^{-d_3 s_{i,2}}, & E_{i,2}^* &= g_4^{s_{i,2}} = g^{d_4 s_{i,2}}, \end{aligned}$$

where  $s_{i,1}, s_{i,2} \in Z_p^*$ . Therefore, algorithm  $\mathcal{C}$  outputs the ciphertext  $\text{CT}^* = (C^*, D^*, E^*, \{(C_i^*, D_{i,1}^*, D_{i,2}^*, E_{i,1}^*, E_{i,2}^*, F_i^*)\}_{i \in [1,l]})$  as required.

- Guess. Algorithm  $\mathcal{A}$  outputs a guess  $\beta'$  for  $\beta$  to guess which game algorithm  $\mathcal{C}$  has been playing, and algorithm  $\mathcal{C}$  forwards  $\beta'$  as its own answer to the  $(q-1)$  assumption.

If  $Z = \hat{e}(g, g)^{s\alpha^{a^{q+1}}}$ , then algorithm  $\mathcal{A}$ 's view of this simulation is identical to the original game, because  $C^* = (M^* || \gamma) \oplus H(Z \cdot \hat{e}(g, g^s)^{\tilde{\alpha}}) = (M^* || \gamma) \oplus H(Z \cdot \hat{e}(g, g)^{\alpha s})$ . On the other hand, if  $Z$  is a random term of  $G_1$ , then all the information about the message  $M^*$  is hidden in the challenge ciphertext. Therefore the advantage of algorithm  $\mathcal{A}$  is 0. As a result, if algorithm  $\mathcal{A}$  distinguishes game  $\text{Game}_0$  from game  $\text{Game}_1$  with a non-negligible probability, then algorithm  $\mathcal{C}$  has a non-negligible advantage in breaking the  $(q-1)$  assumption.

**Lemma 2.** *Assuming that the decisional linear assumption holds in  $G$ , then there is no adversary that distinguishes between the games  $\text{Game}_{j+1}$  and  $\text{Game}_j$  for  $j \in [1, l]$ .*

*Proof.* Assume that there exists an adversary algorithm  $\mathcal{A}$  that can distinguish  $\text{Game}_j$  from  $\text{Game}_{j+1}$ . Then we can build a challenger algorithm  $\mathcal{C}$  that solves the decisional linear assumption.

- Init. Algorithm  $\mathcal{A}$  gives algorithm  $\mathcal{C}$  a challenge access structure  $(\mathbb{M}^*, \rho^*, \{\rho(i)^*\})$ , where  $\mathbb{M}^*$  is an  $l \times n$  matrix.
- Setup. Algorithm  $\mathcal{C}$  randomly chooses  $d_3, d_4, y, \tilde{w}, \tilde{v}, \alpha \in Z_p$ , and computes  $g_3 = g^{d_3}, g_4 = g^{d_4}$ . Then, it sets  $d_1 = x_2, d_2 = x_1$ , and outputs the public parameter as  $pars = (H, g, u, h, w, g_1, g_2, g_3, g_4, \hat{e}(g, g)^\alpha)$  where  $H$  is hash function that maps from  $G_1$  to  $\{0, 1\}^t$  as follows.

$$\begin{aligned} g &= g, & w &= g^{\tilde{w}}, & g_1 &= g^{x_2}, & g_2 &= g^{x_1}, & g_3 &= g^{x_3}, & g_4 &= g^{x_4}, \\ v &= g^{\tilde{v}}, & u &= g^{x_2\alpha}, & h &= g^{-x_2\alpha A_{l'}^*} g^y, & \hat{e}(g, g)^\alpha &= \hat{e}(g, g)^\alpha. \end{aligned}$$

- Phase 1 and Phase 2. To answer an attribute-based private key query on a set of attributes  $\mathbf{A} = \{A_1, \dots, A_k\}$ , algorithm  $\mathcal{C}$  randomly chooses  $r, r', r_1, \dots, r_k, r'_1, \dots, r'_k \in Z_p$ , implicitly sets

$$\begin{aligned} \tilde{r} &= \frac{r\alpha(A_i - A_{l'}^*)}{\alpha(A_i - A_{l'}^*)x_2 + y}, & \tilde{r}' &= r' + \frac{yx_1r}{d_3d_4(\alpha(A_i - A_{l'}^*)x_2 + y)}, \\ r_i &= \frac{\tilde{r}_i(\alpha(A_i - A_{l'}^*)x_2 + y) - \tilde{v}\tilde{r}}{\alpha(A_i - A_{l'}^*)}, & r'_i &= \tilde{r}'_i - \frac{yx_1\tilde{r}_i - x_1\tilde{v}\tilde{r}}{d_3d_4(\alpha(A_i - A_{l'}^*)x_2 + y)}, \end{aligned}$$

and computes

$$\begin{aligned} K_1 &= g^\alpha K_2^{\tilde{w}} = g^\alpha w^{d_1d_2\tilde{r}+d_3d_4\tilde{r}'}, & K_2 &= (g^{x_1})^r g^{r'd_3d_4} = g^{d_1d_2\tilde{r}+d_3d_4\tilde{r}'}, \\ K_{i,1} &= (g^{x_1})^{\alpha(A_i - A_{l'}^*)r_i} = ((u^{A_i}h)^{\tilde{r}_i}v^{-\tilde{r}})^{d_2}, \\ K_{i,2} &= (g^{x_2})^{\alpha(A_i - A_{l'}^*)r_i} = ((u^{A_i}h)^{\tilde{r}_i}v^{-\tilde{r}})^{d_1}, \\ K_{i,4} &= g^{\frac{yx_1r_i - x_1r\tilde{v}}{d_3}} (u^{A_i}h)^{d_4r'_i}(v^{-r'})^{d_4} = ((u^{A_i}h)^{\tilde{r}'_i}v^{-\tilde{r}'})^{d_4}, \\ K_{i,5} &= g^{\frac{yx_1r_i - x_1r\tilde{v}}{d_4}} (u^{A_i}h)^{d_3r'_i}(v^{-r'})^{d_3} = ((u^{A_i}h)^{\tilde{r}'_i}v^{-\tilde{r}'})^{d_3}, \\ K_{i,3} &= (g^{x_1})^{r_i} g^{r'_id_3d_4} = g^{d_1d_2\tilde{r}_i+d_3d_4\tilde{r}'_i}. \end{aligned}$$

- Challenge. Algorithm  $\mathcal{A}$  sends algorithm  $\mathcal{C}$  a message  $M^*$ . Algorithm  $\mathcal{C}$  randomly chooses a vector  $\vec{v} = (\mu, y_2, \dots, y_n) \in Z_p^n$ . Also, for  $i \in [1, l]$  and  $i \neq l$ , algorithm  $\mathcal{C}$  randomly chooses  $\gamma, s_{i,1}, s_{i,2}, z_i \in Z_p, \mu \in Z_p$ . Algorithm  $\mathcal{C}$  implicitly sets  $z_l = x_3 + x_4, s_{l,1} = x_3$ , and computes

$$\begin{aligned} C^* &= H(\hat{e}(g, g)^{\alpha\mu}) \oplus (M^* || \gamma), & D^* &= g^\mu, & E^* &= g^{M^*} h^\gamma, \\ C_l^* &= w^{v_l} Z^{\vec{v}} = w^{v_l} v^{z_l}, & D_{l,1}^* &= g^{x_2x_4} = g_1^{z_l - s_{l,1}}, \\ D_{l,2}^* &= Z^{d_3} g^{-d_3s_{l,2}} = g_3^{z_l - s_{l,2}}, & E_{l,1}^* &= g^{x_1x_3} = g_2^{s_{l,1}}, \\ E_{l,2}^* &= g_4^{s_{l,2}}, & F_l^* &= Z^y = (u^{\rho(l)}h)^{-z_l}, \\ \forall i \neq l \in [1, l] & C_i^* = w^{v_i} v^{z_i}, & D_{i,1}^* &= g_1^{z_i - s_{i,1}}, & E_{i,2}^* &= g_4^{s_{i,2}}, \\ & E_{i,1}^* = g_2^{s_{i,1}}, & D_{i,2}^* &= g_3^{z_i - s_{i,2}}, & F_i^* &= (u^{\rho(i)}h)^{-z_i}, \end{aligned}$$

where  $v_l = \vec{v} \cdot \mathbb{M}_l, v_i = \vec{v} \cdot \mathbb{M}_i, s_{l,2} \in Z_p$ . Therefore, algorithm  $\mathcal{C}$  outputs the ciphertext  $\text{CT}^* = (C^*, D^*, E^*, \{(C_i^*, D_{i,1}^*, D_{i,2}^*, E_{i,1}^*, E_{i,2}^*, F_i^*)\}_{i \in [1, l]})$  as required.

- Guess. Algorithm  $\mathcal{A}$  outputs a guess  $\beta'$  for  $\beta$ .

On the one hand, if  $Z = g^{x_3+x_4}$ , then algorithm  $\mathcal{A}$ 's view of this simulation is identical to the original game. On the other hand, if  $Z$  is randomly chosen from  $G$ , then algorithm  $\mathcal{A}$ 's advantage is nil. Therefore, if algorithm  $\mathcal{A}$  can distinguish game  $\text{Game}_j$  from game  $\text{Game}_{j+1}$  with a non-negligible probability, algorithm  $\mathcal{B}$  has a non-negligible probability in breaking the decisional linear assumption.

**Lemma 3.** *Assuming that the decisional linear assumption holds in  $G$ , then the advantage of an adversary that can distinguish between the games  $\text{Game}_{j+l+1}$  and  $\text{Game}_{j+l}$  for  $j \in [1, l]$  is negligible.*

*Proof.* This proof follows almost the same as that of Lemma 2, except that the simulation is done over the parameters  $g_3$  and  $g_4$  instead of  $g_1$  and  $g_2$ .

This completes the proof of Theorem 1.

#### 4.4 Performance Evaluation and Implementation

Denote  $l$  by the number of attributes in an access structure,  $k$  by the size of an attribute set possessed by each user,  $\chi_1$  by the number of elements in  $I_{\mathbb{M},\rho} = \{\mathcal{I}_1, \dots, \mathcal{I}_{\chi_1}\}$ ,  $\chi_2$  by  $|\mathcal{I}_1| + \dots + |\mathcal{I}_{\chi_1}|$ . Table 2 shows the sizes of the public parameter, the master private key, the ciphertext, the attribute-based private key (i.e., storage complexity) of our expressive CP-ABE scheme supporting partially hidden access structures, where  $|\mathbb{A}|$  is the size of the access structure. Note that our scheme is measured in terms of the number of elements in the prime-order groups. According to the analysis in [12], in terms of the pairing-friendly elliptic curves, prime-order groups have a clear advantage in the parameter sizes over composite-order groups. Table 3 gives the computational costs incurred by the encryption and decryption algorithms in the scheme proposed in this paper. Since regarding the same security level, composite-order groups are several orders of magnitude slower than the prime-order groups [21], and the performance gap will get worse with the increase of security level [9], it is not difficult to see that our expressive CP-ABE scheme with partially hidden access structures in the prime-order groups becomes very competitive.

**Table 2.** The storage overheads in our proposed scheme.

Public parameter	Master private key	Private key	Ciphertext	Group order
11	5	$5k + 3$	$6l + 3 +  \mathbb{A} $	prime

We implement the proposed CP-ABE scheme with partially hidden access structures in Charm [1]<sup>5</sup>, which is a framework developed to facilitate rapid

<sup>5</sup> For the explicit information on Charm, please refer to [1]. Note that since it has been clearly shown in [12, 21] that the efficiency of schemes in composite-order groups is much worse than that of schemes in prime-order groups, we will not implement those schemes in composite-order groups (e.g., [15]).

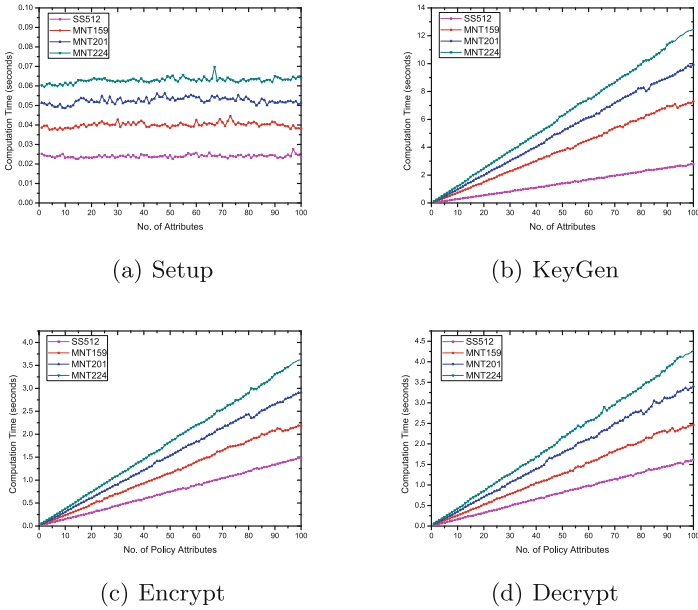


**Table 3.** The computational costs in our proposed scheme, “Expo” and “Multi” denote the exponentiation and multiplication calculation, respectively.

Encrypt			Decrypt		
Multi	Expo	Pairing	Multi	Expo	Pairing
$2l + 1$	$8l + 4$	0	$\leq 5\chi_2 + 2\chi_1$	$\leq \chi_2 + 2\chi_1$	$\leq 6\chi_2 + \chi_1$

prototyping of cryptographic schemes and protocols. Since all Charm routines are designed under the asymmetric groups, our construction is transformed into the asymmetric setting before the implementation. That is, three groups  $G$ ,  $\hat{G}$  and  $G_1$  are used and the pairing  $\hat{e}$  is a function from  $G \times \hat{G}$  to  $G_1$ . Notice that it has been stated in [21] that the assumptions and the security proofs in the symmetric groups can be converted to the asymmetric setting in a generic way. Our experiments are run on a desktop computer with Intel Core i5 – 3470T CPU (4 core 3.20 GHz) and 4 GB RAM running Linux Kernel 3.13.0, which is installed with Charm-0.43 and Python 3.4 for the implementation. Also, we install the PBC library of version 0.5.14 and OpenSSL library of version 1.0.2 for underlying cryptographic operations.

We simulate the algorithms of the proposed scheme over four elliptic curves: SS512 (a symmetric curve with a 512-bit base field), MNT159 (an asymmetric curve with a 159-bit base field), MNT201 (an asymmetric curve with a 201-bit

**Fig. 1.** Performance of our expressive CP-ABE scheme with partially hidden access structures

base field) and MNT224 (an asymmetric curve with a 224-bit base field), which provide security levels of 80-bit, 80-bit, 100-bit and 112-bit, respectively.

In Fig. 1, the performance of the proposed CP-ABE scheme with partially access structures is shown in terms of four algorithms: the setup algorithm Setup (Fig. 1-(a)), the attribute-based private key generation algorithm KeyGen (Fig. 1-(b)), the encryption algorithm Encrypt (Fig. 1-(c)) and the decryption algorithm Decrypt (Fig. 1-(d)). It is not difficult to see from Fig. 1 that SS512 has the best performance, while MNT224 has the most expensive computational cost among all four curves. For each curve, the computation time for the setup algorithm is immutable with the maximum number of attributes allowed in the system, the computation time for the key generation algorithm increases linearly with the size of attribute set, whilst the computation time for the encryption and decryption algorithms grows linearly with the complexity of the access policy. In addition, in our experiments, the computation time of decrypting a ciphertext ranges from 0.30s to 0.80s for a ciphertext with an access policy of 20 attributes and a private key with 20 attributes, and this result is acceptable to most applications in practice.

## 5 Conclusions

A promising solution for preserving data privacy in cloud services is called ciphertext-policy attribute-based encryption (CP-ABE) [22], where data owners upload their data in encrypted forms to the cloud and share them with users with the specified credentials or attributes. In a standard CP-ABE scheme, every ciphertext is attached with an access structure in a cleartext which may leak sensitive information about the recipients and the encrypted message. To address this problem, the notion of CP-ABE with partially hidden access structures [14, 15, 18, 19, 24] was introduced such that the concrete attribute values in access structures are hidden from the public view. Unfortunately, existing CP-ABE schemes with partially hidden access structures [14, 15, 18, 19, 24] either only support restricted access structures or are built in the inefficient composite-order bilinear groups. Motivated by this observation, in this paper, we presented a CP-ABE scheme with partially hidden access structures in the prime-order groups, supporting access structures in monotonic boolean formulas expressed as LSSSs. Also, we formally proved its security, and evaluated its efficiency.

**Acknowledgments.** This research work is supported by the Singapore National Research Foundation under the NCR Award No. NRF2014NCR-NCR001-012.

## References

1. Akinyele, J.A., Garman, C., Miers, I., Pagano, M.W., Rushanan, M., Green, M., Rubin, A.D.: Charm: a framework for rapidly prototyping cryptosystems. *J. Cryptographic Eng.* **3**(2), 111–128 (2013)
2. Beimel, A.: Secure Schemes for Secret Sharing and Key Distribution. Ph.D. thesis, Israel Institute of Technology, Israel Institute of Technology, June 1996

3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (S&P 2007), 20–23, Oakland, California, USA, pp. 321–334. IEEE Computer Society, May 2007
4. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
5. Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
6. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
7. Cheung, L., Newport, C.C.: Provably secure ciphertext policy ABE. In: Proceedings of the ACM Conference on Computer and Communications Security, CCS , Alexandria, Virginia, USA, October 28–31, pp. 456–465. ACM (2007)
8. Fischlin, M., Fischlin, R.: Efficient non-malleable commitment schemes. *J. Cryptology* **24**(1), 203–244 (2011)
9. Freeman, D.M.: Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 44–61. Springer, Heidelberg (2010)
10. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfssdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 579–591. Springer, Heidelberg (2008)
11. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS, Alexandria, VA, USA, October 30 - November 3, vol. 5126. LNCS, pp. 89–98. Springer (2006)
12. Guillevic, A.: Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (eds.) ACNS 2013. LNCS, vol. 7954, pp. 357–372. Springer, Heidelberg (2013)
13. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. *J. Cryptology* **26**(2), 191–224 (2013)
14. Lai, J., Deng, R.H., Li, Y.: Fully secure ciphertext-policy hiding CP-ABE. In: Bao, F., Weng, J. (eds.) ISPEC 2011. LNCS, vol. 6672, pp. 24–39. Springer, Heidelberg (2011)
15. Lai, J., Deng, R.H., Li, Y.: Expressive CP-ABE with partially hidden access structures. In: 7th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2012, pp. 18–19. ACM, Seoul, Korea, May 2–4 2012
16. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 62–91. Springer, Heidelberg (2010)
17. Lewko, A., Waters, B.: Decentralizing attribute-based encryption. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 568–588. Springer, Heidelberg (2011)
18. Li, J., Ren, K., Zhu, B., Wan, Z.: Privacy-aware attribute-based encryption with user accountability. In: Samarati, P., Yung, M., Martinelli, F., Ardagna, C.A. (eds.) ISC 2009. LNCS, vol. 5735, pp. 347–362. Springer, Heidelberg (2009)
19. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. In: Bellare, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 111–129. Springer, Heidelberg (2008)

20. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992)
21. Rouselakis, Y., Waters, B.: Practical constructions and new proof methods for large universe attribute-based encryption. In: ACM SIGSAC Conference on Computer and Communications Security, CCS 2013, pp. 463–474. ACM, Berlin, Germany, November 4–8 2013
22. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EURO-CRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
23. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 53–70. Springer, Heidelberg (2011)
24. Zhang, Y., Chen, X., Li, J., Wong, D.S., Li, H.: Anonymous attribute-based encryption supporting efficient decryption test. In: 8th ACM Symposium on Information, Computer and Communications Security, ASIA CCS 2013, pp. 511–516. ACM, Hangzhou, China - May 08–10 2013

Provable Security

10th International Conference, ProvSec 2016, Nanjing,  
China, November 10-11, 2016, Proceedings

Chen, L.; Han, J. (Eds.)

2016, XIII, 394 p. 34 illus., Softcover

ISBN: 978-3-319-47421-2