Attribute-Based Storage Supporting Secure Deduplication of Encrypted Data in Cloud

Hui Cui, Robert H. Deng, Yingjiu Li, and Guowei Wu

Abstract—Attribute-based encryption (ABE) has been widely used in cloud computing where a data provider outsources his/her encrypted data to a cloud service provider, and can share the data with users possessing specific credentials (or attributes). However, the standard ABE system does not support secure deduplication, which is crucial for eliminating duplicate copies of identical data in order to save storage space and network bandwidth. In this paper, we present an attribute-based storage system with secure deduplication in a hybrid cloud setting, where a private cloud is responsible for duplicate detection and a public cloud manages the storage. Compared with the prior data deduplication systems, our system has two advantages. Firstly, it can be used to confidentially share data with users by specifying access policies rather than sharing decryption keys. Secondly, it achieves the standard notion of semantic security for data confidentiality while existing systems only achieve it by defining a weaker security notion. In addition, we put forth a methodology to modify a ciphertext over one access policy into ciphertexts of the same plaintext but under other access policies without revealing the underlying plaintext.

Index Terms—ABE, Storage, Deduplication.

1 Introduction

Cloud computing greatly facilitates data providers who want to outsource their data to the cloud without disclosing their sensitive data to external parties and would like users with certain credentials to be able to access the data [1], [2], [3], [4], [5]. This requires data to be stored in encrypted forms with access control policies such that no one except users with attributes (or credentials) of specific forms can decrypt the encrypted data. An encryption technique that meets this requirement is called attribute-based encryption (ABE) [6], where a user's private key is associated with an attribute set, a message is encrypted under an access policy (or access structure) over a set of attributes, and a user can decrypt a ciphertext with his/her private key if his/her set of attributes satisfies the access policy associated with this ciphertext. However, the standard ABE system fails to achieve secure deduplication [7], which is a technique to save storage space and network bandwidth by eliminating redundant copies of the encrypted data stored in the cloud. On the other hand, to the best of our knowledge, existing constructions [8], [9], [10], [11] for secure deduplication are not built on attribute-based encryption. Nevertheless, since ABE and secure deduplication have been widely applied in cloud computing, it would be desirable to design a cloud storage system possessing both properties.

We consider the following scenario in the design of an attribute-based storage system supporting secure deduplication of encrypted data in the cloud, in which the cloud will not store a file more than once even though it may receive multiple copies of the same file encrypted under different access policies. A data provider, Bob, intends to upload a file M to the cloud, and share M with users having certain credentials. In order to do so, Bob encrypts M under an access policy $\mathbb{A}$ over a set of attributes, and uploads the corresponding ciphertext to the cloud, such that only users whose sets of attributes satisfying the access policy can decrypt the ciphertext. Later, another data provider, Alice, uploads a ciphertext for the same underlying file M but ascribed to a different access policy $\mathbb{A}'$. Since the file is uploaded in an encrypted form, the cloud is not able to discern that the plaintext corresponding to Alice's ciphertext is the same as that corresponding to Bob's, and will store M twice. Obviously, such duplicated storage wastes storage space and communication bandwidth.

1.1 Our Contributions

In this paper, we present an attribute-based storage system which employs ciphertext-policy attribute-based encryption (CP-ABE) and supports secure deduplication. Our main contributions can be summarized as follows.

- Firstly, the system is the first that achieves the standard notion of semantic security for data confidentiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture [12].
- Secondly, we put forth a methodology to modify a ciphertext over one access policy into ciphertexts of the same plaintext but under any other access policies without revealing the underlying plaintext. This technique might be of independent interest in addition to the application in the proposed storage system.
- Thirdly, we propose an approach based on two cryptographic primitives, including a zero-knowledge proof of knowledge [13] and a commitment scheme [14], to achieve data consistency in the system.

- *Hui Cui is with the Secure Mobile Centre, School of Information Systems, Singapore Management University. E-mail: hcui@smu.edu.sg*
- *Robert H. Deng, Yingjiu Li and Guowei Wu are with the School of Information Systems, Singapore Management University.*

Manuscript received Month Day, 2016; revised Month Day, 2016.

In a typical storage system with secure deduplication (e.g., [9]), to store a file in the cloud, a data provider generates a tag and a ciphertext. The data provider uploads the tag and the ciphertext to the cloud. Upon receiving an outsourcing request from a data provider for uploading a ciphertext and an associated tag, the cloud runs a so-called equality checking algorithm, which checks if the tag in the incoming request is identical to any tags in the storage system. If there is a match, then the underlying plaintext of this incoming ciphertext has already been stored and the new ciphertext is discarded. It is apparent that such a system with a tag appended to the ciphertext does not provide the standard notion of semantic security for data confidentiality [15], because if the plaintexts can be predicated from their tags, an adversary can always make a correct guess by computing the tag of a plaintext and then testing it against the tag in the challenge phase in the semantic security game. To circumvent this obstacle, we bring in our system a hybrid cloud architecture [12], which consists of a private cloud responsible for tag checking and ciphertext regeneration (to be introduced later) and a public cloud storing the ciphertexts. Thanks to this architecture, we manage to achieve semantic security with respect to the public cloud, whilst in terms of the private cloud, a weaker security notion called privacy under chosen distribution attacks (PRV-CDA security) [8] is accomplished under the assumption that the message space is sufficiently large such that each message to be uploaded to the cloud is unpredictable.

However, endowing such a tag checking ability to the private cloud is not sufficient to achieve deduplication in the attribute-based storage system which employs CP-ABE for data encryption. In the proposed attributed-based system, the same file could be encrypted to different ciphertexts associated with different access policies, storing only one ciphertext of the file means that users whose attributes satisfy the access policy of a discarded ciphertext (but not that of the stored ciphertext) will be denied to access the data that they are entitled to. To overcome this problem, we equip the private cloud with another capability named ciphertext regeneration. For a ciphertext c of a plaintext M with access policy $\mathbb{A}$, the private cloud will be provided with a trapdoor key which is generated along with the ciphertext c by a data provider. The private cloud can use the trapdoor key to convert the ciphertext c with access policy $\mathbb{A}$ to a new ciphertext C with another access policy $\mathbb{A}'$ without knowing the underlying message M. Thus, if two data providers happen to upload two ciphertexts corresponding to the same file but under different access policies $\mathbb{A}$ and $\mathbb{A}'$, the private cloud can regenerate a ciphertext for the same underlying file with an access policy $\mathbb{A} \cup \mathbb{A}'$[1] using the corresponding trapdoor key and then store the new ciphertext instead of the old one in the public cloud.

Another key challenge in secure deduplication is to make it secure against duplicate faking attacks [8] in which a legally generated message is unnoticeably replaced by a fake one. In such an attack, a malicious user may intercept an outsourcing request and tamper with the ciphertext, and then sending the modified ciphertext but the original tag to the cloud. Later, an honest data provider wants to upload a ciphertext for an identical file. The cloud spots that the tags of the two ciphertexts match each other, and thus might discard the ciphertext from the honest data provider and keeps the maliciously modified ciphertext. When a user downloads the ciphertext, a tampered message M' rather than the correct M will be returned, which violates data integrity. In order to address this problem, we require the data provider to produce a proof of consistency reflecting that the tag and the ciphertext are legitimately generated. Our approach of producing such a proof makes use of the randomness reuse technique in the generation of the tag and the ciphertext with an additional zero-knowledge proof of knowledge (PoK) [13] on the shared random coin in the tag and the ciphertext. Therefore, it is impossible for an adversary to perform duplicate faking attacks unless the adversary casually obtains the content of the plaintext hidden in the ciphertext.

Unfortunately, the above method only works for the private cloud who is responsible for tag checking. It remains challenging to achieve secure deduplication in the public cloud. Since the public cloud is not involved in any computation or verification, it is indispensable to guarantee that its stored ciphertexts are kept intact without any modification. A straightforward way to achieve this is to save the tags and the ciphertexts in pairs in the public cloud[2], but if the tag and the corresponding ciphertext are both known to the public cloud, then as we mentioned before, it is impossible to obtain semantic security. To achieve the standard security notation for data confidentiality [15], we ask a data provider to generate a label, in addition to the prior tag and ciphertext, using a commitment scheme [14]. This label is bound to the ciphertext and tag using the aforementioned PoK system but reveals no information about the underlying plaintext to the public cloud and users who are not entitled with the decryption privilege, and will be outsourced to the public cloud with the ciphertext instead of the tag, so that even if an adversary who is aware of the data that an honest data provider may upload, the duplicate faking attacks can be detected by users who download and decrypt the data. Note that because the label is stored by the private and public clouds, the tampering behaviour to the label in the public cloud will be immediately detected by the private cloud. Therefore, a user having decryption privilege to the ciphertext can always check the correctness of the plaintext via the label since the tag and the label must be tied to the same plaintext in terms of the proof.

1.2 Related Work

Attribute-Based Encryption. Sahai and Waters [6] introduced the notion of attribute-based encryption (ABE), and then Goyal et al. [16] formulated key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE) as two complimentary forms of ABE. The first KP-ABE construction given in [16] realized the monotonic access structures, the first KP-ABE system supporting the expression of non-monotone formulas was presented in [17] to enable more viable access poli-

1. For simplicity, $\mathbb{A} \cup \mathbb{A}'$ is used to denote an access policy which satisfies both $\mathbb{A}$ and $\mathbb{A}'$.

2. In this way, any user who downloads the file, after decryption, can check the correctness of the decrypted plaintext by matching it to the corresponding tag.

cies, and the first large class KP-ABE system was presented by in the standard model in [18]. Nevertheless, we believe that KP-ABE is less flexible than CP-ABE because the access policy is determined once the user's attribute private key is issued. Bethencourt, Sahai and Waters [19] proposed the first CP-ABE construction, but it is secure under the generic group model. Cheung and Newport [20] presented a CP-ABE scheme that is proved to be secure under the standard model, but it only supports the AND access structures. A CP-ABE system under more advanced access structures is proposed by Goyal et al. [21] based on the number theoretic assumption. In order to overcome the limitation that the size of the attribute space is polynomially bounded in the security parameter and the attributes are fixed ahead, Rouselakis and Waters [22] built a large universe CP-ABE system under the prime-order group. In this paper, the Rouselakis-Waters system is taken as the underlying scheme for the concrete construction.

Secure Deduplication. With the goal of saving storage space for cloud storage services, Douceur et al. [23] proposed the first solution for balancing confidentiality and efficiency in performing deduplication called convergent encryption, where a message is encrypted under a message-derived key so that identical plaintexts are encrypted to the same ciphertexts. In this case, if two users upload the same file, the cloud server can discern the equal ciphertexts and store only one copy of them. Implementations and variants of convergent encryption were deployed in [24], [25], [26], [27], [28]. In order to formalize the precise security definition for convergent encryption, Bellare, Keelveedhi and Ristenpart [8] introduced a cryptographic primitive named message-locked encryption, and detailed several definitions to capture various security requirements. Abadi et al. [9] then strengthened the security definition in [8] by considering the plaintext distributions depending on the public parameters of the schemes. This model was later extended by Bellare and Keelveedhi [11] by providing privacy for messages that are both correlated and dependent on the public system parameters. Since message-locked encryption cannot resist to brute-force attacks where files falling into a known set will be recovered, an architecture that provides secure deduplicated storage resisting brute-force attacks was put forward by Keelveedhi, Bellare and Ristenpart [10] and realized in a system called server-aided encryption for deduplicated storage. In this paper, a similar technique to that in [9] is used to achieve secure deduplication with regard to the private cloud in the concrete construction.

1.3 Organization

The remainder of this paper is organized as follows. In Section 2, we briefly review the notions and definitions to be used in the paper. In Section 3, after depicting the architecture for the attribute-based storage system supporting secure deduplication, we present its security model. We give a concrete attribute-based storage system supporting secure deduplication and analyze its security and performance efficiency in Section 4, and compare it with other related works in the literature in Section 5. We conclude the paper in Section 6.

2 PRELIMINARIES

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

2.1 Bilinear Pairings and Complexity Assumptions

Suppose that Groupgen is a probabilistic polynomial-time algorithm that inputs a security parameter λ, and outputs a triplet $(G,\ p,\ g)$ where G is a group of order p that is generated from g, and p is a prime number. We define $\hat{e}: G \times G \rightarrow G_1$ to be a bilinear map if it has the following properties [29].

- Bilinear: 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: $\hat{e}(g, g) \neq 1$.

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 [22]. The decisional $(q-1)$ problem is that for any probabilistic polynomial-time algorithm, given $\overrightarrow{y} =$

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

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

2.2 Symmetric Encryption

A symmetric encryption (SE) scheme $\mathcal{SE}$ with a key space $\mathcal{K}$ and a message space $\mathcal{M}$ [30] is composed of two algorithms: an encryption algorithm $\mathcal{SE}.\mathrm{Enc}(K,\ m)$ which outputs a ciphertext CT on input a key $K \in \mathcal{K}$ and a message $m \in \mathcal{M}$, and a decryption algorithm $\mathcal{SE}.\mathrm{Dec}(K, \mathrm{CT})$ which outputs a message m or a failure symbol $\perp$ on input a key $K \in \mathcal{K}$ and a ciphertext CT.

Let st be the state information. A symmetric encryption scheme $\mathcal{SE}$ is secure under chosen plaintext attacks (IND-CPA secure), if for any PPT adversary $\mathcal{A} = (\mathcal{A}_1,\ \mathcal{A}_2)$, the advantage function

$$
\mathbf{Adv}_{\mathcal{SE},\mathcal{A}}^{\text{IND-CPA}}(\lambda) = \Pr\left[b' = b \;\middle|\; \begin{array}{l} K \leftarrow \mathcal{K}; b \leftarrow \{0,1\} \\ (m_0, m_1, st) \leftarrow \mathcal{A}_1(1^{\lambda}) \\ \mathrm{CT}^* \leftarrow \mathcal{SE}.\mathrm{Enc}(K, m_b) \\ b' \leftarrow \mathcal{A}_2(par, m_0, m_1, st, \mathrm{CT}^*) \end{array} \right] - 1/2
$$

is negligible in the security parameter λ, where $|m_0| = |m_1|$.

2.3 Commitment Scheme

A commitment scheme $\mathcal{CME}$ is composed of the following three algorithms [14]: parameter generation algorithm CPG which takes a security parameter λ as input and outputs the public parameters $cpars$, committal algorithm Com which takes the public parameters $cpars$ and data x as input and outputs a commitment com to x along with a decommittal key dec, and deterministic verification algorithm Ver which takes the public parameter $cpars$, data x, a commitment com and a decommittal key dec as input and outputs 1 to indicate that it accepts or 0 to indicate that it rejects.

A commitment scheme should be both binding which means that the decommit phase can successfully open to only one value, and hiding which means that the commit phase does not reveal any information about x. For X $\in$ {Hiding, Binding}, the advantages

$$\mathbf{Adv}^{\mathrm{X}}_{\mathcal{CMT},\mathcal{A}}(\lambda) = 2 \cdot \Pr[\mathrm{X}^{\mathcal{A}}_{\mathcal{CMT}} \Rightarrow \text{true}] - 1$$

referring to the games of the hiding and binding properties in Fig. 1 are negligible in the security parameter λ.

2.4 Access Structures and Linear Secret Sharing Schemes

We review the the notions of access structures and linear secret sharing schemes in [31], [32] as follows.

Definition 1. (Access Structures). Let $\{P_1, ..., P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,...,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \subseteq \mathbb{A}$. An (monotone) access structure is a (monotone) collection $\mathbb{A}$ of non-empty subsets of $\{P_1, ..., P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,...,P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

Definition 2. (Linear Secret Sharing Schemes). Let P be a set of parties. Let $\mathbb{M}$ be a matrix of size $l \times n$. Let $\rho : \{1, ..., l\} \rightarrow P$ be a function that maps a row to a party for labeling. Let p be a prime number. A secret sharing scheme Π over a set of parties P is a linear secret-sharing scheme (LSSS) 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 Π. For $i = 1, ..., l$, the i-th row of matrix $\mathbb{M}$ is labeled by a party $\rho(i)$, where $\rho : \{1, ..., l\} \rightarrow P$ is a function that maps a row to a party for labeling. Considering that the column vector $v = (\mu, r_2, ..., r_n)$, where $s \in Z_p$ is the secret to be shared and $r_2, ..., r_n \in Z_p$ are randomly chosen, then $\mathbb{M}v$ is the vector of l shares of the secret s according to Π. The share $(\mathbb{M}v)_i$ belongs to party $\rho(i)$.

It has been noted in [31] that every LSSS enjoys the linear reconstruction property. Denote Π as an LSSS for access structure $\mathbb{A}$. Let $\mathbf{A}$ be an authorized set, and define $I \subseteq \{1, ..., l\}$ as $I = \{i | \rho(i) \in \mathbf{A}\}$. Then the vector $(1, 0, ..., 0)$ 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 s according to Π, 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 $\mathbb{M}$ [33].

Boolean Formulas [31]. Access structures 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 system architecture and the formal definition of ciphertext-policy attribute-based storage system supporting secure deduplication.

3.1 System Architecture

The architecture of our attribute-based storage system with secure deduplication is shown in Fig. 2 in which four entities are involved: data providers, attribute authority (AA), cloud and users. A data provider wants to outsource his/her data to the cloud and share it with users possessing certain credentials. The AA issues every user a decryption key associated with his/her set of attributes. The cloud consists of a public cloud which is in charge of data storage and a private cloud which performs certain computation such as tag checking. When sending a file storage request, each data provider firstly creates a tag T and a label L associated with the data, and then encrypts the data under an access structure over a set of attributes. Also, each data provider generates a proof *pf* on the relationship of the tag T, the label L and the encrypted message ct[3], but this proof will not be stored anywhere in the cloud and is only used during the checking phase for any newly generated storage request. After receiving a storage request, the private cloud first checks the validity of the proof *pf*, and then tests the equality of the new tag T with existing tags in the system. If there is no match for this new tag T, the private cloud adds the tag T and the label L to a tag-label list, and forwards the label and the encrypted data, (L, ct) to the public cloud for storage. Otherwise, let ct' be the ciphertext whose tag matches the new tag and L' be the label associated with ct', and then the private cloud executes as follows.

- If the access policy in ct is a subset of that in ct', the private cloud simply discards the new storage request; else, if the access policy in ct' is a subset of that in ct, the private cloud asks the public cloud to replace the stored pair (L', ct') with the new pair (L, ct) where $L = L'$.
- If the access policies in ct and ct' are not mutually contained, the private cloud runs the ciphertext re-generation algorithm to yield a new ciphertext for the same underlying plaintext file and associated with an access structure which is the union of the two access

3. To keep the notation succinct, we use c to denote the combination of the encrypted data and the corresponding access structure.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2656120, IEEE Transactions on Big Data

proc Initialize	proc Initialize
$cpars \leftarrow \text{CPG}(1^\lambda); b \in \{0,1\}$	$cpars \leftarrow \text{CPG}(1^\lambda)$
Return $cpars$	Return $cpars$
proc LR(x_0, x_1)	proc Finalize(com, x_0, dec_0, x_1, dec_1)
$(com, dec) \leftarrow \text{Com}(cpars, x_b)$	$d_0 \leftarrow \text{Ver}(cpars, x_0, com, dec_0)$
Return com	$d_1 \leftarrow \text{Ver}(cpars, x_1, com, dec_1)$
proc Finalize(b')	Return $(x_0 \neq x_1 \wedge d_0 = 1 \wedge d_1 = 1)$
Return $(b' = b)$	

Fig. 1: Game Hiding$_{\mathcal{CMT}}$ (left) achieves the hiding property and Game Binding$_{\mathcal{CMT}}$ (right) achieves the binding property. Note that LR can only be called once.

structures, and forwards the original label and the resulting ciphertext to the public cloud.

At the user side, each user can download an item, and decrypt the ciphertext with the attribute-based private key generated by the AA if this user's attribute set satisfies the access structure. Each user checks the correctness of the decrypted message using the label, and accepts the message if it is consistent with the label.

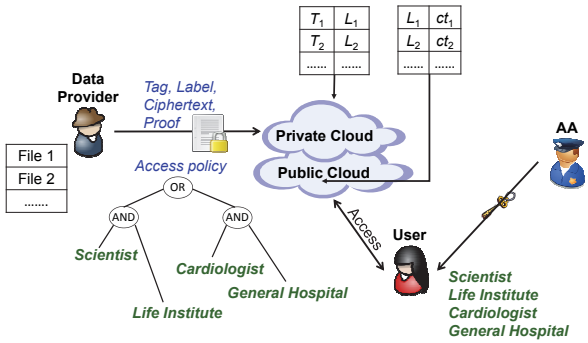


Fig. 2: System architecture of attribute-based storage with secure deduplication.

Concerning the adversarial model of our storage system, we assume that the private cloud is "curious-but-honest" such that it will attempt to obtain the encrypted messages but it will honestly follow the protocols, whereas the public cloud is distrusted such that it might tamper with the label and ciphertext pairs outsourced from the private cloud (note that such a misbehaviour will be detected by either the private cloud or the user via the accompanied label). Another difference between the private cloud and the public cloud is that the former can not collude with users[4], but the latter could collude with users. This assumption is in line with the real world practice where the private cloud is trusted more than the public cloud. We assume that data users may try to access data beyond their authorized privileges. In addition to trying to obtain plaintext data from the cloud, malicious outsiders may also commit duplicate faking attacks as described before.

4. Otherwise, the private cloud can regenerate the ciphertext under an access policy that an unprivileged user can satisfy, thereby obtaining the hidden plaintext and breaking the security of the storage system.

3.2 Framework

Our ciphertext-policy attribute-based storage system with secure deduplication consists of the following algorithms: setup algorithm Setup, attribute-based private key generation algorithm KeyGen, encryption algorithm Encrypt, validity testing algorithm Validity-Test, equality testing algorithm Equality-Test, re-encryption algorithm Re-encrypt and decryption algorithm Decrypt.

- Setup$(1^\lambda) \rightarrow (pars, msk)$. Taking the security parameter λ as the input, this setup algorithm outputs the public parameter $pars$ and the master private key msk for the system.
  This algorithm is run by the AA.
- KeyGen$(pars, msk, \mathbf{A}) \rightarrow sk_{\mathbf{A}}$. Taking the public parameter $pars$, the master private key msk and an attribute set $\mathbf{A}$ as the input, this attribute-based private key generation algorithm generates an attribute-based private key $sk_{\mathbf{A}}$ for the attribute set $\mathbf{A}$.
  This algorithm is run by the AA.
- Encrypt$(pars, M, \mathbb{A}) \rightarrow (sk_T, \text{CT})$. Taking the public parameter $pars$, a message M and an access structure $\mathbb{A}$ over the universe of attributes as the input, this encryption algorithm outputs a trapdoor key sk_T and a tuple CT $= (T, L, ct, pf)$, where T and L are the tag and the label associated with M respectively, ct is the ciphertext which includes the encryption of M as well as the access structure $\mathbb{A}$, and pf is a proof on the relationship of tag T, label L and ciphertext ct.
  This algorithm is run by the data provider. Both sk_T and CT are forwarded to the private cloud. Note that sk_T can not be disclosed to any third party, so it must be sent to the private cloud in a secure manner.
- Validity-Test$(pars, \text{CT}) \rightarrow 1/0$. Taking the public parameter $pars$ and a tuple CT as the input, this validity testing algorithm parses CT as (T, L, ct, pf), and outputs 1 if pf is a valid proof for (T, L, ct) or 0 otherwise.
  This algorithm is run by the private cloud.
- Equality-Test$(pars, (T_1, L_1, ct_1), (T_2, L_2, ct_2)) \rightarrow 1/0$. Taking the public parameter $pars$ and two tuples (T_1, L_1, ct_1) and (T_2, L_2, ct_2) as the input, this equality testing algorithm outputs 1 if both (T_1, L_1, ct_1), (T_2, L_2, ct_2) are generated from the same underlying message or 0 otherwise.
  This algorithm is run by the private cloud.

- Re-encrypt($pars$, sk_T, (L, ct), $\mathbb{A}'$) $\rightarrow (L, ct')$. Taking the public parameter $pars$, the trapdoor key sk_T, a tag and ciphertext pair (L, ct) and an access structure $\mathbb{A}'$ as the input, this re-encryption algorithm outputs a new ciphertext ct' associated with $\mathbb{A}'$ sharing the same label L of the ciphertext ct'.
  This algorithm is run by the private cloud.
- Decrypt($pars$, (L, ct), $\mathbf{A}$, $sk_{\mathbf{A}}$) $\rightarrow M/\perp$. Taking the public parameter $pars$, a label and ciphertext pair (L, ct) and an attribute-based private key $sk_{\mathbf{A}}$ associated to an attribute set $\mathbf{A}$ as the input, this decryption algorithm outputs either the message M when the private key $sk_{\mathbf{A}}$ satisfies the access structure of the ciphertext ct and the label L is consistent with M (to be defined later), or a symbol $\perp$ indicating the failure of the decryption.
  This algorithm is run by the user.

We require that a ciphertext-policy attribute-based storage system with secure deduplication Π is correct, meaning that the decryption algorithm correctly decrypts a ciphertext of an access structure $\mathbb{A}$ with an attribute-based private key on $\mathbf{A}$, when $\mathbf{A}$ is an authorized set of $\mathbb{A}$. Formally, for all messages M, and all attribute sets $\mathbf{A}$ and access structures $\mathbb{A}$ with authorized $\mathbf{A}$ satisfying $\mathbb{A}$, if $(pars, msk) \leftarrow$ Setup(1^λ), $sk_{\mathbf{A}} \leftarrow$ KeyGen($pars$, msk, $\mathbf{A}$), $(sk_T, \text{CT}) \leftarrow$ Encrypt($pars$, M, $\mathbb{A}$), $1 \leftarrow$ Validity-Test($pars$, CT), then Decrypt($pars$, (L, ct), $\mathbf{A}$, $sk_{\mathbf{A}}$) $= M$. Additionally, for all messages M, we require that if $(sk_T, \text{CT}) \leftarrow$ Encrypt($pars$, M, $\mathbb{A}$), $1 \leftarrow$ Validity-Test($pars$, CT), and $(sk'_T, \text{CT}') \leftarrow$ Encrypt($pars$, M, $\mathbb{A}'$), $1 \leftarrow$ Validity-Test($pars$, CT$'$), then Equality-Test($pars$, (T, L, ct), $(T', L', ct')) = 1$.

Notice that with respect to a concrete construction, the input $\mathbb{A}$ of the encryption algorithm Encrypt will be set to be the corresponding policy $(\mathbb{M}, \rho)$.

3.3 Security Definitions

Traditionally, an encryption system is required to provide privacy of the encrypted data, which is captured by indistinguishability under either chosen plaintext attacks (IND-CPA) or chosen ciphertext attacks (IND-CCA). However, neither IND-CPA nor IND-CCA is feasible in an encrypted storage system with secure deduplication, since it can be easily broken by an adversary in either IND-CPA or IND-CCA security game as follows. An adversary, given a challenge CT* for a plaintext m_b with $b \in \{0, 1\}$ where m_0, m_1 are chosen by the adversary, can output the correct b by creating a tag T for m_b and running the equality testing algorithm to see whether T matches the tag T^* of CT*. Noticeably, it is impossible to design an encryption scheme with an equality-checking tag to satisfy the standard notions of confidentiality [9]. Thus, we alternatively aim to achieve IND-CPA security at the public cloud side, whilst preserving a security notion called PRV-CDA security (privacy under chosen distribution attacks) [8] at the private cloud side under the assumption that the message space $\mathcal{M}(\lambda)$ is sufficiently large such that the plaintexts in the system are unpredictable (i.e., given the public parameter and encryption of a randomly selected plaintext in the message space $\mathcal{M}(\lambda)$, it is infeasible for any polynominal time algorithm $\mathcal{A}$ to obtain the plaintext).

IND-CPA Security. Denote our attribute-based storage system with secure deduplication Π. The definition of selective IND-CPA security with respect to the public cloud in Π is given in Fig. 3, where we restrain algorithm $\mathcal{A}$ to issuing queries to the key generation oracle on attribute sets satisfying the access structures $\mathbb{A}_0$ and $\mathbb{A}_1$.

An attribute-based storage system with secure deduplication Π is IND-CPA secure if the advantage function referring to the security game $\text{Game}^{\text{IND}}_{\Pi,\mathcal{A}}$

$$\mathbf{Adv}^{\text{IND}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[b' = b]$$

is negligible in the security parameter λ for any probabilistic polynomial-time (PPT) adversary algorithm $\mathcal{A}$.

PRV-CDA Security. Based on the definition of PRV-CDA given in [8], the definition of PRV-CDA for Π is shown in Fig. 3, where the adversary is given an additional trapdoor key for the challenge ciphertext but is not given access to any attribute-based private keys (as the private cloud is not allowed to collude with users).

An attribute-based storage system with secure deduplication Π is PRV-CDA secure if the advantage function referring to the security game $\text{Game}^{\text{PRV}}_{\Pi,\mathcal{A}}$

$$\mathbf{Adv}^{\text{PRV-CDA}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[b' = b]$$

is negligible in the security parameter λ for any PPT adversary algorithm $\mathcal{A}$.

With regard to a storage system, it is crucial to ensure consistency [9] to resist duplicate faking attacks such that a legitimate message will not be unnoticeably replaced by a fake one. Consistency in our attribute-based storage system with secure deduplication can be divided into ciphertext consistency, tag and label consistency. Ciphertext consistency guarantees that given a ciphertext outsourced by an honest data provider, an adversary who has no idea about the encrypted data can not generate another valid ciphertext with the same tag but under a different plaintext to cheat the private cloud. Tag/Label consistency ensures consistency of the data used in the tag/label derivation and the ciphertext generation such that an adversary is not able to create a tag/label that does not match the underlying data to cheat a user having access to the encrypted data.

Consistency. Ciphertext consistency for our attribute-based storage system with secure deduplication Π is given in Fig. 4, in which given a ciphertext (T, L, ct, pf) and the public parameter, an adversary wins the game if it outputs another ciphertext (T, L', ct', pf') such that pf' is valid for (T, L', ct'). This game prevents an adversary from capturing an outsourcing request from an honest data provider and replacing the corresponding ciphertext to another ciphertext without being detected by the private cloud. Taking the definition for consistency in [9] into consideration, we depict the security game for tag/label consistency for our system Π in Fig. 4, which provides security against duplicate faking attacks where a legitimate message is replaced by a fake one without being discovered. Specifically, assume that an adversary creates and uploads a ciphertext ct' of M' associated with a tag and label pair for M. Later, an honest data provider, holding M computes and uploads the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2656120, IEEE Transactions on Big Data

Security game for selective IND-CPA: $\text{Game}^{\text{IND}}_{\Pi,\mathcal{A}}$	Security game for PRV-CDA: $\text{Game}^{\text{PRV}}_{\Pi,\mathcal{A}}$
$(pars, msk) \leftarrow \text{Setup}(1^\lambda)$; $b \leftarrow \{0,1\}$	$(pars, msk) \leftarrow \text{Setup}(1^\lambda)$
$(\text{st}, \mathbb{A}_0, \mathbb{A}_1, M_0, M_1) \leftarrow \mathcal{A}^{\text{KeyGen}_{msk}(\cdot)}(pars)$	$(M_0^*, M_1^*) \leftarrow \mathcal{M}(\lambda)$
$(sk_T, \text{CT}) \leftarrow \text{Encrypt}(pars, M_b, \mathbb{A}_0)$	$(\text{st}, \mathbb{A}^*) \leftarrow \mathcal{A}(pars)$
$(L, ct^*) \leftarrow \text{Re-encrypt}(pars, sk_T, (L, ct), \mathbb{A}_1)$	$(sk_T^*, \text{CT}^*) \leftarrow \text{Encrypt}(pars, M_b^*, \mathbb{A}^*)$
$b' \leftarrow \mathcal{A}^{\text{KeyGen}_{msk}(\cdot)}(pars, \text{st}, M_0, M_1, (L, ct^*))$	$b' \leftarrow \mathcal{A}(pars, \text{st}, sk_T^*, \text{CT}^*)$
Return $b' = b$	Return $b' = b$

Fig. 3: Security game for selective IND-CPA (left) and PRV-CDA (right), where st is information collected by the adversary.

Ciphertext-Consistency security game: $\text{Game}^{\text{CC}}_{\Pi,\mathcal{A}}$	Tag (or Label)-Consistency security game: $\text{Game}^{\text{TC (or LC)}}_{\Pi,\mathcal{A}}$
$pars \leftarrow \text{Setup}(1^\lambda)$	$pars \leftarrow \text{Setup}(1^\lambda)$
$\text{CT} \leftarrow \text{Encrypt}(pars, M, \mathbb{A})$	$(M, \text{CT}) \leftarrow \mathcal{A}(pars)$
$\text{CT}' \leftarrow \mathcal{A}(pars, \text{CT})$	If $M = \perp$ or $\text{CT} = \perp$ Return false
$M' \leftarrow \text{Decrypt}(pars, (L', ct'), \mathbf{A}, sk_{\mathbf{A}})$	$M' \leftarrow \text{Decrypt}(pars, (L', ct'), \mathbf{A}, sk_{\mathbf{A}})$
If $1 \leftarrow \text{Validity-Test}(pars, \text{CT}')$	$\text{CT}' \leftarrow \text{Encrypt}(pars, M, \mathbb{A})$
$\wedge\ (M \neq M') \wedge (\text{CT} \cap \text{CT}' = T)$	If $1 \leftarrow \text{Equality-Test}(pars, (L, T, ct), (L', T', ct'))$
Return true	$\wedge\ (M \neq M')$
	Return true

Fig. 4: Security games for consistency.

encryption ct of M. Since the tags of ct and ct' are equal, the private cloud continues to ask the public cloud to store only ct'. Later, the honest data provider, who expects to recover M, downloads and decrypts ct', but it obtains M' instead of M. In addition, the duplicate faking attacks can occur when an adversary tampers with the label and ciphertext pairs stored in the public cloud by modifying (L, ct) to (L', ct'). Note that any misbehaviour to the label in the public cloud will be easily spotted by the private cloud due to that each label is stored in both public and private clouds, and thus the tampering to the ciphertext will be found by those who can decrypt it via checking whether the label derived from the decryption matches the given label.

An attribute-based storage system for secure deduplication is consistent if the advantage function referring to the security game $\text{Game}^{\text{XC}}_{\Pi,\mathcal{A}}$ for XC $\in$ {CC, TC, LC}

$$\mathbf{Adv}^{\text{XC}}_{\Pi,\mathcal{A}}(\lambda) \stackrel{\text{def}}{=} \Pr[\text{Game}^{\text{XC}}_{\Pi,\mathcal{A}} \Rightarrow \text{true}]$$

is negligible in the security parameter λ for any PPT adversary algorithm $\mathcal{A}$.

4 Attribute-Based Storage with Secure Deduplication

In this section, we describe a concrete construction of an attribute-based storage system supporting secure deduplication, analyze its security, and show its performance from theoretical and experimental analysis.

4.1 Construction

Let $\mathcal{SE} = (\mathcal{SE}.\text{Enc}, \mathcal{SE}.\text{Dec})$ be a symmetric encryption scheme with a message space $\mathcal{M}$ and a key space $\mathcal{K}$. On the basis of the large universe CP-ABE scheme proposed in [22], below we present an attribute-based storage system with secure deduplication.

- Setup. This algorithm takes the security parameter λ as the input. It randomly chooses a group G of a prime order p with a generator g, and a bilinear pairing $\hat{e} : G \times G \rightarrow G_1$. Then, it randomly chooses collision resistant hash functions $f_0 : G_1 \rightarrow Z_p$, $f_1 : \mathcal{M} \rightarrow Z_p$, $F : G_1 \rightarrow \mathcal{K}$, $H : G^5 \rightarrow Z_p$. Also, it randomly chooses $\alpha \in Z_p^*$, $u, h, v, w \in G$. The public parameter is $pars = (f_0, f_1, F, H, g, u, h, w, v, \hat{e}(g,g)^\alpha)$, and the master private key is $msk = g^\alpha$.
- KeyGen. This algorithm takes the public parameter $pars$, the master private key msk and a set $\mathbf{A} = \{A_1, ..., A_{|\mathbf{A}|}\}$ of attributes as the input. It randomly chooses $r, r_1, ..., r_{|\mathbf{A}|} \in Z_p^*$, and computes

$$sk_1' = g^\alpha w^r, \quad sk_2' = g^r,$$
$$\forall i \in \mathbf{A} \ \ sk_2^{(i)} = g^{r_i}, \quad sk_1^{(i)} = (u^{A_i} h)^{r_i} v^{-r}.$$

  It outputs the attribute-based private key $sk_{\mathbf{A}} = (sk_1', \{sk_1^{(i)}\}_{i\in\mathbf{A}}, sk_2', \{sk_2^{(i)}\}_{i\in\mathbf{A}})$ associated with a set of attributes $\mathbf{A}$.
- Encrypt. This algorithm takes the public parameter $pars$, a message $M \in \mathcal{M}$ and an LSSS access structure $(\mathbb{M}, \rho)$ where ρ is a function which 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 $\overrightarrow{v} = (\mu^5, y_2, ..., y_n) \in Z_p^n$, of which the values will be used to share the encryption exponent μ. For $i = 1, ..., l$, it calculates $v_i = \overrightarrow{v} \cdot \mathbb{M}_i$, where $\mathbb{M}_i$ is the vector corresponding to the i-th row of the matrix

5. In addition, if μ is set to be $\mathcal{H}(\beta, M)$ where $\mathcal{H}$ is a hash function mapping the input to an element from Z_p^*, then the proposed scheme can achieve the IND-CCA security in the random oracle model, which is the generic transformation technology from IND-CPA security to IND-CCA security proposed in [30].

$\mathbb{M}$. In addition, it randomly chooses $\beta \in G_1$, z_1, ..., $z_l \in Z_p$, and computes

$$U = g^{f(M)\mu},\ L = g^{f_1(M)} h^{f_0(\beta)},$$
$$E = \mathcal{SE}.\text{Enc}(F(\beta), M)$$
$$B = g^{\mu},\ C = \beta \cdot \hat{e}(g,g)^{\alpha\mu},$$
$$\forall i \in [1,l]\ C_i = w^{v_i} v^{z_i},\ D_i = g^{z_i},\ E_i = (u^{\rho(i)} h)^{-z_i},$$
$$\text{PoK}\{(M,\beta) : U = B^{f(M)} \wedge L = g^{f_1(M)} h^{f_0(\beta)}\}.$$

It outputs a trapdoor key $sk_T = w^{\mu}$, and a tuple of tag, label, ciphertext and proof CT = (T, L, ct, pf) where $T = (U, B)$, $ct = \big((\mathbb{M}, \rho), E, B, C, \{(C_i, D_i, E_i)\}_{i\in[1,l]}\big)$, and pf is a zero-knowledge proof of knowledge (PoK) for the equality of μ in U, B and $f(M)$ in U, L without leaking the values of μ, M and β. Here PoK is a zero-knowledge proof composed of $(U, B, L, \theta_1, \theta_2)$ and can be computed as follows. It randomly chooses $d_1, d_2 \in Z_p^*$, and computes

$$R_1 = B^{d_1}, \quad R_2 = g^{d_1} h^{d_2}, \quad c = H(U, B, L, R_1, R_2),$$
$$\theta_1 = d_1 - c \cdot f_1(M), \quad \theta_2 = d_2 - c \cdot f_0(\beta).$$

Note that according to the binding property of the commitment scheme [14], each L can only be obtained from a unique pair of M and β, which guarantee the consistency of the ciphertext stored by the public cloud.

- Validity-Test. This algorithm takes the public parameter $pars$ and a ciphertext CT as the input. To test the validity of the ciphertext, it computes

$$R_1 = U^c B^{\theta_1}, \quad R_2 = L^c g^{\theta_1} h^{\theta_2}.$$

If $c = H(U, B, L, R_1, R_2)$, it accepts CT, and stores $\big(L, ((\mathbb{M}, \rho), E, B, C, \{C_i, D_i, E_i\}_{i\in[1,l]})\big)$ to the public cloud. Otherwise, it rejects CT.
- Equality-Test. This algorithm takes the public parameter $pars$ and two tags (U_1, B_1) and (U_2, B_2) of the outsourced data as input. It outputs 1 if $\hat{e}(U_1, B_2) = \hat{e}(U_2, B_1)$. Otherwise, it outputs 0.
- Re-encrypt. This algorithm takes the public parameter $pars$, a trapdoor key sk_T, a ciphertext $\big((\mathbb{M}, \rho), E, B, C, \{(C_i, D_i, E_i)\}\big)$ with a label L and an LSSS access structure $(\mathbb{M}', \rho')$ where the function ρ' associates the rows of $\mathbb{M}'$ to attributes as the input. Let $\mathbb{M}'$ be an $l' \times n'$ matrix. It randomly chooses $\overrightarrow{v} = (\bar{\mu}, \bar{y}_2', ..., \bar{y}_{n'}') \in Z_p^{n'}$. For each row $\mathbb{M}_{i'}' = (m_{i'1}', ..., m_{i'n'}')$ of $\mathbb{M}'$ where $i' \in [1, l']$, it randomly chooses $z_{i'}' \in Z_p$. Let $\overrightarrow{v}' = (\mu', \bar{y}_2', ..., \bar{y}_{n'}')$ for $\mu' = \mu + \bar{\mu}$. For $i' \in [1, l']$, it outputs the new ciphertext as

$$B' = B \cdot g^{\bar{\mu}},\ L' = L,\ E' = E,\ C' = C \cdot \hat{e}(g,g)^{\alpha\bar{\mu}},$$
$$C_{i'}' = w^{\mathbb{M}_{i'}' \overrightarrow{v}'} v^{z_{i'}'},\ D_{i'}' = g^{z_{i'}'},\ E_{i'}' = (u^{\rho'(i')} h)^{-z_{i'}'},$$

where $C_{i'}'$ can be computed as follows without knowing the values of μ and $\widetilde{\mu}$.

$$C_{i'}' = w^{\mathbb{M}_{i'}' \overrightarrow{v}'} v^{z_{i'}'} = w^{(\mu' m_{i'1}' + ... + y_n' m_{i'n'}')} v^{z_i'}$$
$$= w^{\mu m_{i1}'} w^{(\bar{\mu} m_{i'1}' + ... + \bar{y}_n' m_{i'n'}')} v^{z_{i'}'}.$$

It is straightforward to see that the distribution of $\big(L', ((\mathbb{M}', \rho'), E', B', C', \{C_i', D_i', E_i'\}_{i'\in[1,l']})\big)$ is consistent with that outputted by the encryption algorithm Encrypt($pars$, M, $(\mathbb{M}', \rho')$).
- Decrypt. This algorithm takes the public parameter $pars$, a ciphertext $\big((\mathbb{M}, \rho), E, B, C, \{C_i, D_i, E_i\}_{i\in[1,l]}\big)$ with the corresponding label L and a private key $sk_{\mathbf{A}}$ for an attribute set $\mathbf{A}$ as the input. Suppose that an attribute set $\mathbf{A}$ satisfies the access structure $(\mathbb{M}, \rho)$. Define I as $I = \{i : \rho(i) \in \mathbf{A}\}$. Denote by $\{w_i \in Z_p\}_{i\in I}$ a set of constants such that if $\{v_i\}$ are valid shares of any secret μ according to $(\mathbb{M}, \rho)$, then $\sum_{i\in I} w_i v_i = \mu$. It computes the message M as

$$\frac{\hat{e}(B, sk_1')}{\prod_{i\in I}(\hat{e}(C_i, sk_2')\hat{e}(D_i, sk_1^{(i)})\hat{e}(E_i, sk_2^{(i)}))^{w_i}}$$
$$= \frac{\hat{e}(g,g)^{\alpha\mu}\hat{e}(g,w)^{\mu r}}{\prod_{i\in I}\hat{e}(g,w)^{r v_i w_i}} = \hat{e}(g,g)^{\alpha\mu},$$

and cancels out $\hat{e}(g,g)^{\alpha\mu}$ from C to obtain β. Then, it computes $M = \mathcal{SE}.\text{Dec}(F(\beta), E)$. If $g^{f_1(M)} h^{f_0(\beta)} = L$, it outputs M. Otherwise, it outputs a failure symbol $\perp$.

Correctness. The correctness for the decryption algorithm follows that of the original attribute-based encryption scheme in [22]. The correctness for the validity testing algorithm relies on the zero-knowledge proof of knowledge system PoK, which is straightforward. The correct of equality testing algorithm is guaranteed by the properties of groups equipped with bilinear maps. If $T_1 = (U_1, B_1)$ and $T_2 = (U_2, B_2)$ are created by the encryption scheme on the same underlying message M, then

$$\hat{e}(U_1, B_2) = \hat{e}(g^{f_1(M)\mu_1}, g^{\mu_2}) = \hat{e}(g,g)^{f_1(M)\mu_1\mu_2},$$
$$\hat{e}(U_2, B_1) = \hat{e}(g^{f_1(M)\mu_2}, g^{\mu_1}) = \hat{e}(g,g)^{f_1(M)\mu_1\mu_2}.$$

Thus, $\hat{e}(U_1, B_2) = \hat{e}(U_2, B_1)$ as required.

Remarks. Note that a similar idea for ciphertext regeneration has been put forward by Lai et al. [34], but in their method, the trapdoor key is created by the AA and can be used to transform any ciphertext over one access policy into those ciphertexts of an identical plaintext under other access policies. Whereas in our system, we resort to a one (trapdoor key) to one (ciphertext) framework such that even one trapdoor key is compromised, the system is still secure for other ciphertexts.

4.2 Security

We begin with proving the security of the zero-knowledge proof of knowledge used in the proposed construction, which plays an important role in proving the security of the proposed storage system.

Lemma 1. The PoK is a secure zero-knowledge proof of knowledge system of witness (M, β).

Proof. Since the *completeness* of PoK is straightforward, we focus on its *soundness* and *zero-knowledge*.

Soundness. Assume there are two transcripts with the same (U, L) but different challenges c', c and different responses (θ_1', θ_2') and (θ_1, θ_2).

Then (μ, M) can be extracted from

$$U = B^{f_1(M)} = B^{\frac{\theta_1' - \theta_1}{c - c'}},$$
$$L = g^{f_1(M)} h^{f_0(\beta)} = g^{\frac{\theta_1' - \theta_1}{c - c'}} h^{\frac{\theta_2' - \theta_2}{c - c'}}.$$

Zero-knowledge. The simulator randomly chooses $\theta_1, \theta_2 \in Z_p^*$, $c \in Z_p^*$, and computes

$$R_1 = U^c B^{\theta_1}, \quad R_2 = L^c g^{\theta_1} h^{\theta_2}.$$

Then it sets $c = H(U, B, L, R_1, R_2)$. □

Next, we prove that the proposed storage system preserves the privacy of the encrypted data in terms of public cloud and private cloud, respectively.

Theorem 1. Assuming that the $(q-1)$ assumption holds in G, $\mathcal{SE}$ is a secure symmetric encryption scheme and L is generated following a secure commitment scheme, then the proposed attribute-based storage system with secure deduplication is selectively indistinguishable regarding the view of the public cloud.

Proof. The Rouselakis-Waters scheme [22] is known to be selectively indistinguishable assuming that the $(q-1)$ assumption holds in G. Our proof for Theorem 1 mostly follows that in [22] except that in the challenge phase, E^* and $L^* = g^{f_1(M_b^*)} h^{f_0(\beta)}$ will be added to the original challenge ciphertext. Note that E^* will not disclose any information about M_b^* due to the security of the underlying $\mathcal{SE}$ scheme, and L^* will not tell any information about M_b^* due to the security of the underlying commitment scheme. □

Theorem 2. Assuming that the decisional $(q-1)$ assumption holds in G, the decisional BDH assumption holds in G, $\mathcal{SE}$ is a secure symmetric encryption scheme and PoK is a secure zero-knowledge proof of knowledge, then the proposed attribute-based storage system with secure deduplication is PRV-CDA secure.

Proof. The PRV-CDA security is composed of the security of encryption (adversary $\mathcal{A}_1$) and re-encryption (adversary $\mathcal{A}_2$) algorithms. The security against the adversary algorithm $\mathcal{A}_1$ is twofold: the ciphertext and the proof. In terms of the ciphertext, the proof follows that in [22] except that in the challenge phase, E^* and L^* (computed as that in Theorem 1) will be added to the challenge ciphertext. Concerning the proof, due to the property of zero-knowledge proof of knowledge, it discloses no information about M_b^*.

Below we describe the security proof for the adversary algorithm $\mathcal{A}_2$ under the decisional BDH assumption. Suppose that there exists an adversary algorithm $\mathcal{A}_2$ that breaks the PRV-CDA security of our system. Then we can build a challenger algorithm $\mathcal{B}$ that solves the decisional BDH problem. Algorithm $\mathcal{B}$ is given (g, g^a, g^b, g^c, Z), and its goal is to output 1 if $Z = \hat{e}(g,g)^{abc}$ and 0 if Z is uniform in G_T.

Algorithm $\mathcal{B}$ randomly chooses $x \in Z_p^*$, $u, h, v \in G$, and computes $w = g^x$. It sets the public parameter as $pars = (f, H, g, u, h, w, v, \hat{e}(g^a, g^b))$ where f, H are collision resistant hash functions. This implies that the master private key $\alpha = ab$ is unknown to algorithm $\mathcal{B}$.

When algorithm $\mathcal{A}_2$ outputs an access structure $(\mathbb{M}^*, \rho^*)$, algorithm $\mathcal{B}$ firstly chooses a plaintext $M_b^* \in \{M_0^*, M_1^*\}$ $(b \in \{0,1\})$ from the message space, and then it randomly chooses $\widetilde{c}, y_2, ..., y_n \in Z_p$, and sets $\overrightarrow{v} = (c, y_2, ..., y_n)$, $\widetilde{\overrightarrow{v}} = (\widetilde{c}, y_2, ..., y_n)$. Also, algorithm $\mathcal{B}$ randomly chooses $\beta \in G_1, z_1, ..., z_l \in Z_p$. It outputs the trapdoor key, tag and ciphertext tuple as

$$sk_T^* = w^c = (g^c)^x, \ \ L^* = g^{f_1(M_b^*)} h^{f_0(\beta)},$$
$$E^* = \mathcal{SE}.\mathrm{Enc}(F(\beta), M_b^*),$$
$$B^* = g^c, \ C^* = \beta^* \cdot Z, \ \widetilde{B}^* = g^{\widetilde{c}}, \ \widetilde{C}^* = \beta \cdot Z,$$
$$C_i^* = w^{\mathbb{M}_i^* \overrightarrow{v}} v^{z_i}, \ D_1^* = g^{z_i}, \ E_i^* = (u^{\rho^*(i)} h)^{-z_i},$$

where for $i \in [1, l]$, C_i^* can be computed as follows without knowing the value of c.

$$\begin{aligned} C_i^* &= w^{\mathbb{M}_i^* \overrightarrow{v}} v^{z_i} = w^{(cm_{i1}^* + ... + y_n m_{in}^*)} v^{z_i} \\ &= (w^c)^{m_{i1}^*} w^{(y_2 m_{i2}^* + ... + y_n m_{in}^*)} v^{z_i} \\ &= (g^c)^{x m_{i1}^*} w^{(y_2 m_{i2}^* + ... + y_n m_{in}^*)} v^{z_i}. \end{aligned}$$

Since $Z = \hat{e}(g,g)^{abc} = \hat{e}(g^a, g^b)^c$, it is straightforward that the distribution of $(L^*, ((\mathbb{M}^*, \rho^*), E^*, B^*, C^*, \{C_i^*, D_i^*, E_i^*\}))$ and sk_T^* are the same as the input of the re-encryption algorithm in the view of algorithm $\mathcal{A}_2$.

Finally, algorithm $\mathcal{A}_2$ outputs a guess b'. If $b' = b$, algorithm $\mathcal{B}$ outputs 1 meaning $Z = \hat{e}(g,g)^{abc}$. Otherwise, it outputs 0.

When $Z = \hat{e}(g,g)^{abc}$, E^* is created using a secure $\mathcal{SE}$ scheme, and L^* is generated using a secure commitment scheme, the perspective of algorithm $\mathcal{A}_2$ is the same as that in the real game. When Z is uniform in G_T, E^* and L^* are randomly generated, the value of b is information-theoretically hidden from algorithm $\mathcal{A}_2$. Therefore, if algorithm $\mathcal{A}_2$ breaks the PRV-CDA security of the above scheme, algorithm $\mathcal{B}$ solves the decisional BDH problem, or breaks the security of the underlying $\mathcal{SE}$ scheme, or breaks the security of the commitment scheme. □

Finally, we prove that the proposed storage system supports secure deduplication.

Theorem 3. Assume that PoK is a secure zero-knowledge proof of knowledge and L is generated following a secure commitment scheme. Then the attribute-based storage system with secure deduplication is consistent.

Proof. Based on the property of zero-knowledge proof of knowledge, it is straightforward to see that our attribute-based storage system for secure deduplication is ciphertext consistent. Thus, it remains to prove that the system is tag consistent. The tag L in our scheme is constructed using a commitment scheme [14]. Thus, if an adversary breaks the tag consistency of the above system, then this adversary can be used to break the security for the underlying commitment scheme of which the security has been analyzed in [14]. □

4.3 Performance Evaluation

Recall that our attribute-based storage system is built upon the ciphertext-policy attribute-based encryption scheme proposed by Rouselakis and Waters [22] which could not resist duplication behaviours. Let $|pars|$, $|msk|$, $|ct|$, $|L|$, $|T|$, $|sk|$, $|\mathbb{A}|$ be the sizes of the public parameter, the master

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2656120, IEEE Transactions on Big Data

TABLE 1: Comparison of storage complexity between the based scheme [22] and our storage system.

	System public parameter $\|pars\|$	System master private key $\|msk\|$	Public Cloud label and ciphertext $\|ct\| + \|L\|$	Private Cloud tag and label $\|T\| + \|L\|$	User private key $\|sk\|$
CP-ABE [22]	6	1	$3l + 2 + \|\mathbb{A}\|$	–	$2k + 2$
The proposed storage system	10	1	$3l + 5 + \|\mathbb{A}\|$	3	$2k + 2$

private key, the ciphertext, the label, the tag, the decryption key and the access structure, respectively. Denote l by the number of attributes in an access structure, and k by the size of an attribute set ascribed to a user's credentials. Table 1 compares the storage complexity of our system with that in [22]. It is clear that our system is efficient in terms of the introduced storage overhead, which adds the underlying CP-ABE scheme [22] 4 elements to the system public parameter and 3 element to the ciphertext stored by the public cloud, with an additional private cloud storing 3 elements.

Let l be the number of attributes presented in an access structure, and k be the size of an attribute set associated with the private key. Denote y by the number of existing tags stored by the private cloud. Table 2 shows the number of exponential and paring operations in our storage system. For example, it requires at most $k + 2$ exponential operations and $3k + 1$ paring operations to decrypt a ciphertext. Table 3 compares the computational costs incurred at the data provider, the cloud, and the user for one file storage between the system in [22] and our system. It is not difficult to see that the computational requirement for the user in our system is almost twice that in the underlying CP-ABE scheme in [22]. With regard to the data provider, it requires 4 extra exponential operations resulted from the tag, label, proof and trapdoor key in addition to the computational cost of the underlying scheme in [22] lacking the capability of secure deduplication. In terms of the private cloud, our solution takes $5 + (6l + 2)$ exponential operations and $2y$ pairing operations, among which 5 exponential operations are used to check the validity of the proof, $6l+2$ exponential operations are related to the ciphertext regeneration if necessary[6] and $2y$ pairing operations are calculated to check whether the plaintext hidden in the outsourcing request has existed in the public cloud.

TABLE 3: Comparison of computational costs between the underlying scheme [22] and our storage system.

		Data Provider	Private Cloud	User
CP-ABE [22]	Expo	$5l + 2$	–	$\leq k$
	Pairing	0	–	$\leq 3k + 1$
Our storage system	Expo	$5l + 6$	$5 + (6l + 2)$	$\leq k + 2$
	Pairing	0	$2y$	$\leq 3k + 1$

4.4 Implementation

We implement the algorithms of our storage system in Charm [35][7], which is a framework developed to facilitate rapid prototyping of cryptographic schemes and protocols. Since all Charm routines are designed under the asymmetric groups, our construction is transformed to 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 [22] that the assumptions and the security proofs can be converted to the asymmetric setting in a generic way. We use the Charm-0.43 and the Python 3.4 in our implementation. Along with the Charm-0.43, we install the PBC library for the underlying cryptographic operations. Our experiments are run on a laptop with Intel Core i5-4210U CPU @ 1.70GHz and 4.00 GB RAM running 64-bit Ubuntu 16.04.

We simulate the proposed attribute-based storage system with secure deduplication over four different elliptic curves: SS512, MNT159, MNT201 and MNT224, where SS512 is a supersingular elliptic curve with the symmetric Type 1 pairing on it, and the pairings on the other three curves are asymmetric Type 3 pairings. These four curves provide the security level of 80-bit, 80-bit, 100-bit and 112-bit, respectively. Fig. 5 shows the computation complexity of the proposed attribute-based storage system supporting secure deduplication in terms of four algorithms: key generation algorithm KeyGen (Fig. 5-(a)), encryption algorithm Encrypt (Fig. 5-(b)), re-encryption algorithm Re-encrypt (Fig. 5-(c)) and decryption algorithm Decrypt (Fig. 5-(d)). As illustrated in Fig. 5, SS512 has the best performance, while MNT224 has the most expensive computational cost among all the curves. For each curve, the average computation time of key generation increases linearly with the size of attributes set whilst the average computation time of encryption and re-encryption grows linearly with the complexity of the access policy. In terms of the four curves used in our experiments, the average computation time of decrypting a ciphertext ranges from 1.60s to 5.80s for a ciphertext with 100 attributes using a private key with 100 attributes. Clearly, the proposed attribute-based storage system with secure deduplication is sufficiently efficient to be applied in practice.

5 Discussion

In this section, we provide further elaboration on the two main techniques we introduced in this paper.

5.1 Adaptable Attribute-Based Encryption

Lai et al. [34] presented a cryptographic primitive called adaptable CP-ABE, where a semi-trusted proxy is introduced into the setting of CP-ABE. The proxy, given a system wide trapdoor key, is able to transform any ciphertext under one access policy into ciphertexts of the same plaintext

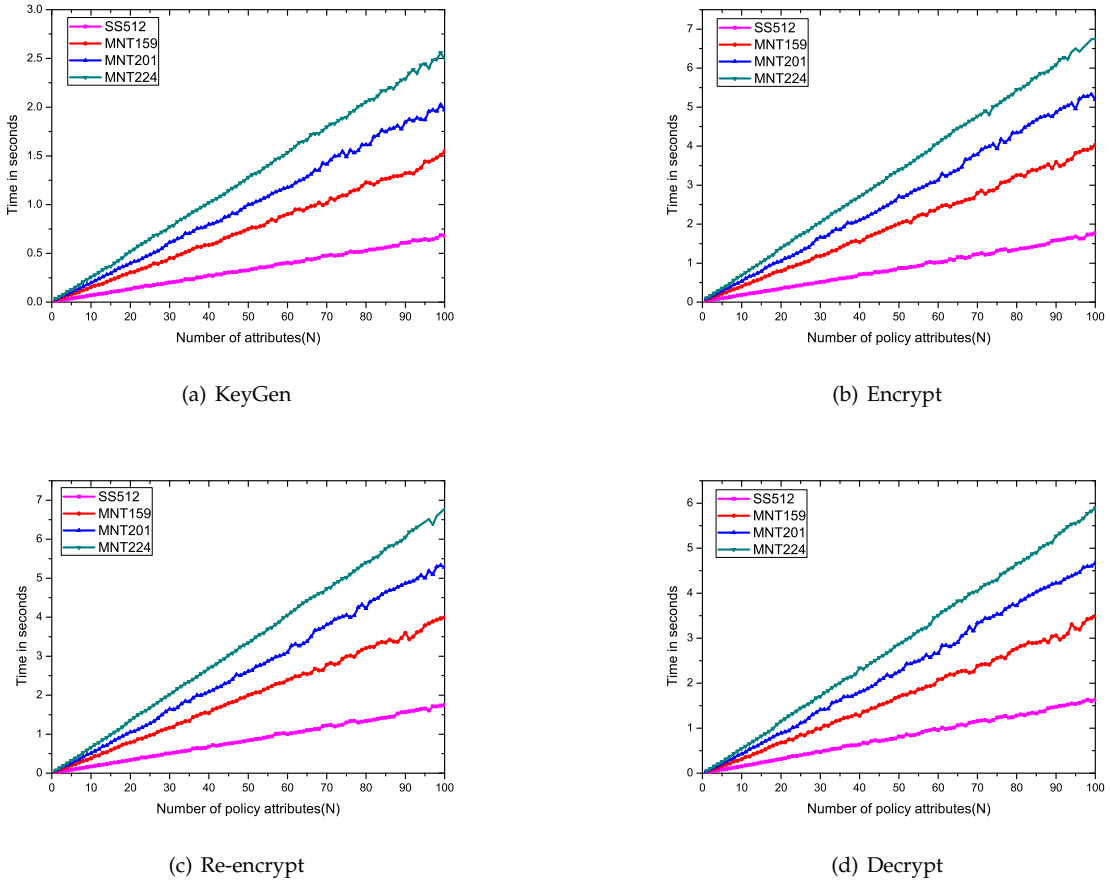
6. Recall that ciphertext regeneration is only executed when the access structures associated with the incoming and existing ciphertexts are not mutually compatible.

7. For the explicit information on Charm, please refer to [35].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2656120, IEEE Transactions on Big Data

TABLE 2: Computational overheads in our storage system.

	Tag	Label	Encrypt	Proof	Trapdoor key	Re-encrypt	Validity	Equality	Decrypt
Expo	2	2	$5l+1$	3	1	$6l+2$	5	0	$\leq k+2$
Pairing	0	0	0	0	0	0	0	$2y$	$\leq 3k+1$



(a) KeyGen (b) Encrypt

(c) Re-encrypt (d) Decrypt

Fig. 5: Performance of our attribute-based storage system supporting secure deduplication.

under any other access policies without learning any information about the plaintext during the process of transformation. However, this method of using a single trapdoor key for all ciphertexts is quite risky, since if the single key is compromised, the security for the system will be totally broken. An adversarial user using the compromised trapdoor key can regenerate a ciphertext into an access structure that his/her attributes satisfy, and thus he/she can obtain the plaintext not intended for him/her. Besides, the trapdoor key in [34] is generated by the AA who already controls the decryption keys in the system, so it is desirable to reduce its power in manipulating the encryption. Unlike that in [34], our technique is one-to-one such that each trapdoor key can only be used to transform its corresponding ciphertext. Therefore, even at some point, a trapdoor key is comprised, the damage is limited to one message. At a high level, our technique brings another way to build adaptive CP-ABE systems from a different point of view.

5.2 Deduplication in Hybrid Cloud

An inherent drawback of the existing approaches to achieve secure deduplication (e.g., [8], [23]) is that they cannot satisfy the standard security definition for confidentiality such as semantic security (See Section 3.3 for the reason). To solve this problem, a weaker security notion called privacy under chosen-distribution attacks [8] was put forward under the assumption that the input message is sufficiently unpredictable. Different from the existing method of defining a weaker security notion for the cloud storage system with secure deduplication, a hybrid cloud architecture, consisting of a pair of public and private clouds, is introduced in our storage system such that the semantic security becomes achievable for the public cloud. This framework of twin clouds has been widely adopted in practice, where the security of the public cloud usually confronts more challenges than that of the private cloud, and hence it is desirable to have stronger data confidentiality protection at the public cloud side. We believe that the hybrid cloud architecture is a promising approach to storage systems with deduplication,

in which the encrypted data is outsourced to the public cloud whilst the deduplication checking is handled by the private cloud.

6 Conclusions

Attribute-based encryption (ABE) has been widely used in cloud computing where data providers outsource their encrypted data to the cloud and can share the data with users possessing specified credentials. On the other hand, deduplication is an important technique to save the storage space and network bandwidth, which eliminates duplicate copies of identical data. However, the standard ABE systems do not support secure deduplication, which makes them costly to be applied in some commercial storage services. In this paper, we presented a novel approach to realize an attribute-based storage system supporting secure deduplication. Our storage system is built under a hybrid cloud architecture, where a private cloud manipulates the computation and a public cloud manages the storage. The private cloud is provided with a trapdoor key associated with the corresponding ciphertext, with which it can transfer the ciphertext over one access policy into ciphertexts of the same plaintext under any other access policies without being aware of the underlying plaintext. After receiving a storage request, the private cloud first checks the validity of the uploaded item through the attached proof. If the proof is valid, the private cloud runs a tag matching algorithm to see whether the same data underlying the ciphertext has been stored. If so, whenever it is necessary, it regenerates the ciphertext into a ciphertext of the same plaintext over an access policy which is the union set of both access policies. The proposed storage system enjoys two major advantages. Firstly, it can be used to confidentially share data with other users by specifying an access policy rather than sharing the decryption key. Secondly, it achieves the standard notion of semantic security while existing deduplication schemes only achieve it under a weaker security notion.

Acknowledgments

This research work is supported by the Singapore National Research Foundation under the NCR Award Number NRF2014NCR-NCR001-012.

References

[1] D. Quick, B. Martini, and K. R. Choo, *Cloud Storage Forensics*. Syngress Publishing / Elsevier, 2014. [Online]. Available: http://www.elsevier.com/books/cloud-storage-forensics/quick/978-0-12-419970-5

[2] K. R. Choo, J. Domingo-Ferrer, and L. Zhang, "Cloud cryptography: Theory, practice and future research directions," *Future Generation Comp. Syst.*, vol. 62, pp. 51–53, 2016.

[3] K. R. Choo, M. Herman, M. Iorga, and B. Martini, "Cloud forensics: State-of-the-art and future directions," *Digital Investigation*, vol. 18, pp. 77–78, 2016.

[4] Y. Yang, H. Zhu, H. Lu, J. Weng, Y. Zhang, and K. R. Choo, "Cloud based data sharing with fine-grained proxy re-encryption," *Pervasive and Mobile Computing*, vol. 28, pp. 122–134, 2016.

[5] D. Quick and K. R. Choo, "Google drive: Forensic analysis of data remnants," *J. Network and Computer Applications*, vol. 40, pp. 179–193, 2014.

[6] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, ser. Lecture Notes in Computer Science, vol. 3494. Springer, 2005, pp. 457–473.

[7] B. Zhu, K. Li, and R. H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system," in *6th USENIX Conference on File and Storage Technologies, FAST 2008, February 26-29, 2008, San Jose, CA, USA*. USENIX, 2008, pp. 269–282.

[8] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, ser. Lecture Notes in Computer Science, vol. 7881. Springer, 2013, pp. 296–312.

[9] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," in *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 8042. Springer, 2013, pp. 374–391.

[10] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Server-aided encryption for deduplicated storage," in *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013*. USENIX Association, 2013, pp. 179–194.

[11] M. Bellare and S. Keelveedhi, "Interactive message-locked encryption and secure deduplication," in *Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings*, ser. Lecture Notes in Computer Science, vol. 9020. Springer, 2015, pp. 516–538.

[12] S. Bugiel, S. Nürnberger, A. Sadeghi, and T. Schneider, "Twin clouds: Secure cloud computing with low latency - (full version)," in *Communications and Multimedia Security, 12th IFIP TC 6 / TC 11 International Conference, CMS 2011, Ghent, Belgium, October 19-21,2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 7025. Springer, 2011, pp. 32–44.

[13] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems (extended abstract)," in *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*. ACM, 1985, pp. 291–304.

[14] M. Fischlin and R. Fischlin, "Efficient non-malleable commitment schemes," in *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, ser. Lecture Notes in Computer Science, vol. 1880. Springer, 2000, pp. 413–431.

[15] S. Goldwasser and S. Micali, "Probabilistic encryption," *J. Comput. Syst. Sci.*, vol. 28, no. 2, pp. 270–299, 1984.

[16] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, ser. Lecture Notes in Computer Science, vol. 5126. Springer, 2006, pp. 89–98.

[17] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*. ACM, 2007, pp. 195–203.

[18] A. B. Lewko and B. Waters, "Unbounded HIBE and attribute-based encryption," in *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6632. Springer, 2011, pp. 547–567.

[19] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*. IEEE Computer Society, 2007, pp. 321–334.

[20] L. Cheung and C. C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*. ACM, 2007, pp. 456–465.

[21] V. Goyal, A. Jain, O. Pandey, and A. Sahai, "Bounded ciphertext policy attribute based encryption," in *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik,*

Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations, ser. Lecture Notes in Computer Science, vol. 5126. Springer, 2008, pp. 579–591.

[22] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013*. ACM, 2013, pp. 463–474.

[23] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in *ICDCS*, 2002, pp. 617–624.

[24] M. W. Storer, K. M. Greenan, D. D. E. Long, and E. L. Miller, "Secure data deduplication," in *Proceedings of the 2008 ACM Workshop On Storage Security And Survivability, StorageSS 2008, Alexandria, VA, USA, October 31, 2008*. ACM, 2008, pp. 1–10.

[25] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de-duplication," in *Uncovering the Secrets of System Administration: Proceedings of the 24th Large Installation System Administration Conference, LISA 2010, San Jose, CA, USA, November 7-12, 2010*. USENIX Association, 2010.

[26] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui, "A secure cloud backup system with assured deletion and version control," in *2011 International Conference on Parallel Processing Workshops, ICPPW 2011, Taipei, Taiwan, Sept. 13-16, 2011*. IEEE Computer Society, 2011, pp. 160–167.

[27] P. Puzio, R. Molva, M. Önen, and S. Loureiro, "Cloudedup: Secure deduplication with encrypted data for cloud storage," in *IEEE 5th International Conference on Cloud Computing Technology and Science, CloudCom 2013, Bristol, United Kingdom, December 2-5, 2013, Volume 1*. IEEE Computer Society, 2013, pp. 363–370.

[28] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage," in *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 8437. Springer, 2014, pp. 99–118.

[29] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *CRYPTO*, ser. Lecture Notes in Computer Science, vol. 2139. Springer-Verlag, 2001, pp. 213–219.

[30] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," *J. Cryptology*, vol. 26, no. 1, pp. 80–101, 2013.

[31] A. B. Lewko and B. Waters, "Decentralizing attribute-based encryption," in *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6632. Springer, 2011, pp. 568–588.

[32] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography, Taormina, Italy, March 6-9, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6571. Springer, 2011, pp. 53–70.

[33] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Israel Institute of Technology, Israel Institute of Technology, June 1996.

[34] J. Lai, R. H. Deng, Y. Yang, and J. Weng, "Adaptable ciphertext-policy attribute-based encryption," in *Pairing-Based Cryptography - Pairing 2013 - 6th International Conference, Beijing, China, November 22-24, 2013, Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 8365. Springer, 2013, pp. 199–214.

[35] J. A. Akinyele, C. Garman, I. Miers, M. W. Pagano, M. Rushanan, M. Green, and A. D. Rubin, "Charm: a framework for rapidly prototyping cryptosystems," *J. Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

Hui Cui received her Ph.D. degree in the School of Computing and Information Technology, University of Wollongong, Australia. She is currently a postdoctoral research fellow in the Secure Mobile Centre under the School of Information Systems, Singapore Management University, Singapore. Her research interests include cryptography, applied cryptography, cloud security and so on.

Robert H. Deng has been a Professor at the School of Information Systems, Singapore Management University since 2004. Prior to this, he was Principal Scientist and Manager of Infocomm Security Department, Institute for Infocomm Research, Singapore. His research interests include data security and privacy, multimedia security, network and system security. He has served/is serving on the editorial boards of many international journals in security, such as IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, the International Journal of Information Security, and IEEE Security and Privacy Magazine. He is the chair of the Steering Committee of the ACM Asia Conference on Computer and Communications Security (ASIACCS). He received the University Outstanding Researcher Award from the National University of Singapore in 1999 and the Lee Kuan Yew Fellow for Research Excellence from the Singapore Management University in 2006. He was named Community Service Star and Showcased Senior Information Security Professional by (ISC)2 under its Asia-Pacific Information Security Leadership Achievements program in 2010. He is the Fellow of IEEE.

Yingjiu Li is currently an Associate Professor in the School of Information Systems at Singapore Management University (SMU). His research interests include RFID Security and Privacy, Mobile and System Security, Applied Cryptography and Cloud Security, and Data Application Security and Privacy. He has published over 130 technical papers in international conferences and journals, and served in the program committees for over 80 international conferences and workshops. Yingjiu Li is a senior member of the ACM and a member of the IEEE Computer Society. The URL for his web page is http://www.mysmu.edu/faculty/yjli/.

Guowei Wu is a master student in the Department of Computer Science, Jinan University, Guangzhou. He is also a visiting student in the School of Information Systems, Singapore Management University, Singapore.