Attribute-Based Encryption With Efficient Verifiable Outsourced Decryption

Baodong Qin, Robert H. Deng, Shengli Liu, and Siqi Ma

***Abstract*— Attribute-based encryption (ABE) with outsourced decryption not only enables fine-grained sharing of encrypted data, but also overcomes the efficiency drawback (in terms of ciphertext size and decryption cost) of the standard ABE schemes. In particular, an ABE scheme with outsourced decryption allows a third party (e.g., a cloud server) to transform an ABE ciphertext into a (short) El Gamal-type ciphertext using a public transformation key provided by a user so that the latter can be decrypted much more efficiently than the former by the user. However, a shortcoming of the original outsourced ABE scheme is that the correctness of the cloud server's transformation cannot be verified by the user. That is, an end user could be cheated into accepting a wrong or maliciously transformed output. In this paper, we first formalize a security model of ABE with verifiable outsourced decryption by introducing a verification key in the output of the encryption algorithm. Then, we present an approach to convert any ABE scheme with outsourced decryption into an ABE scheme with verifiable outsourced decryption. The new approach is simple, general, and almost optimal. Compared with the original outsourced ABE, our verifiable outsourced ABE neither increases the user's and the cloud server's computation costs except some nondominant operations (e.g., hash computations), nor expands the ciphertext size except adding a hash value (which is <20 byte for 80-bit security level). We show a concrete construction based on Green *et al.*'s ciphertext-policy ABE scheme with outsourced decryption, and provide a detailed performance evaluation to demonstrate the advantages of our approach.**

***Index Terms*— Attributed-based encryption, data sharing, decryption outsourcing, verifiability.**

I. Introduction

TRADITIONALLY, access controls to data operate on the assumption that data servers can be trusted to keep data confidential and enforce access control policies correctly. However, this assumption is no longer true today since services are increasingly storing data across many servers that are shared with other data owners. An example of this is cloud data storage where cloud service providers are not in the same trusted domains as end users, and hardware platforms are not under the direct control of data owners. To mitigate users' privacy concerns about their data, a common solution is to store data in encrypted form so that it will remain private, even if data servers or storage devices are not trusted or compromised. The encrypted data, however, must be amenable to sharing and access control.

Data encryption using symmetric or public key cryptography is not amenable to scalable access control. A promising approach to address this issue is attribute-based encryption (ABE), first proposed by Sahai and Waters [1]. ABE schemes can be divided into two categories: Ciphertext-Policy ABE (CP-ABE) and Key-Policy ABE (KP-ABE) [2], depending on the access policy is embedded into the ciphertext or the user's private key. In CP-ABE, an access policy $\mathbb{A}$ is embedded in a ciphertext CT and a user's private key SK is associated with a set S of attributes. The ciphertext CT can be decrypted by SK if and only if $f(\mathbb{A}, S) = 1$ for some predicate function f, meaning that $S \in \mathbb{A}$. In KP-ABE, every ciphertext is associated with a set of attributes, and every user's private key is associated with an access policy on attributes. A user is able to decrypt a ciphertext only if the set of attributes associated with the ciphertext satisfies the access policy associated with the user's private key. Both CP-ABE and KP-ABE can prevent any unauthorized users from accessing data, even if the user stores data in an untrusted server. Such properties of ABE schemes are very attractive in the area of cloud data storage.

However, a drawback of the standard ABE schemes is their relatively large ciphertext size and high decryption cost, and this problem is especially acute for resource limited devices such as mobile devices. Specifically, in an ABE scheme, the size of the ciphertext and the cost of decryption grow with the complexity of the access structures/policies. Moreover, current constructions of ABE schemes, see [2]–[5], rely on pairing-based groups and require many pairing operations (which are usually more expensive than exponentiations) in decryption. Though there exist ABE schemes with constant ciphertext size and/or constant number of pairing operations in decryption, their access structures are restricted to AND gates or threshold gates [6], [7], which severely limit their practical applications. To overcome this problem, Green et al. [8] suggested to outsource decryption in attribute-based encryption. In their approach, shown in Fig. 1, a user's private key is

Manuscript received July 6, 2014; revised February 15, 2015; accepted March 1, 2015. Date of publication March 4, 2015; date of current version May 15, 2015. This work was supported in part by the Office of Research, Singapore Management University, under Grant 14-C220-SMU-006, in part by the Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20110073110016, and in part by the Scientific Innovation Projects of Shanghai Education Committee under Grant 12ZZ021. The work of S. Liu was supported by the National Natural Science Foundation of China under Grant 61170229 and Grant 61373153. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shantanu D. Rane. *(Corresponding author: Shengli Liu.)*

B. Qin is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, the Singapore Management University, Singapore 188065, and also with the Southwest University of Science and Technology, Mianyang 621010, China (e-mail: qinbaodong@sjtu.edu.cn).

R. H. Deng and S. Ma are with the School of Information Systems, Singapore Management University, Singapore 188065 (e-mail: robertdeng@smu.edu.sg; siqi.ma.2013@phdis.smu.edu.sg).

S. Liu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: slliu@sjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2015.2410137

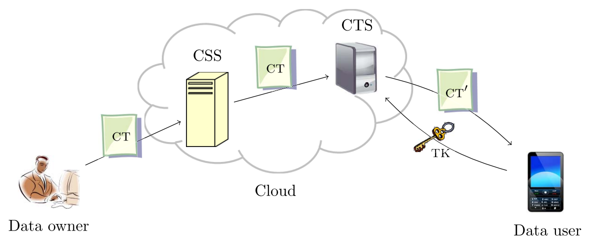


Fig. 1. System model for ABE with outsourced decryption.

split into a "transformation key" (denoted by TK), and an El Gamal-type secret key (denoted by DK). The transformation key can be publicly shared with a proxy, called Ciphertext Transformation Server (CTS), while the secret key DK must be kept private by the user. ABE ciphertexts are stored in a Cloud Storage Server (CSS). A ciphertext CT stored in the CSS is first submitted to the CTS which uses the key TK to transform CT into a simple and short El Gamal-type ciphertext CT$'$ of the same message, instead of being decrypted by the user directly. From CT$'$, the user is able to recover the message using the secret key DK with just one exponentiation operation. The user's transformation key can transform any ABE ciphertext satisfied by user's attributes, without revealing any information of the underlying message to a malicious CTS. Thus, the user saves both bandwidth and local computation time significantly. In the following, we will use the term ABE with outsourced decryption and the term outsourced ABE interchangeably.

In some application scenarios, it is necessary to check the correctness of the transformation. As explained in [9], to save computation time, a lazy proxy may return a ciphertext CT$'$ it transformed previously for the same user or a malicious proxy may return a transformation of another (or modified) ciphertext to the user. Some of the recent verifiable computation techniques [10]–[12] could be leveraged to construct ABE schemes with verifiable outsourced decryption. However, they are currently impractical for ABE systems. As explained in [8], the solutions in [10] and [11] rely on Gentry's fully homomorphic encryption system [13] and one "bootstrapping" operation of the homomorphic operation would take about 30 minutes for a high security parameter [14]. The solutions in [12] allow a client to outsource pairing operations to a server. However, the client still needs to compute multiple exponentiations in the target group for every pairing it outsources. To make their outsourced ABE system verifiable, Green et al. [8] proposed a simple method to adapt their RCCA (replayable chosen-ciphertext attack) systems to such a setting. They appended a tag $H(r)$ to the ciphertext, where r is the ciphertext randomness and H is a hash function modeled as a random oracle. In their RCCA secure construction, since the final decryption allows recovery of the encryption randomness r, the user can compute $H(r)$ and check whether it matches the tag. This approach essentially requires the original tag be untampered with. Otherwise, a malicious CTS could replace the original ciphertext and its tag with a new ciphertext and a new corresponding tag, and then transform the new ciphertext using the user's transformation key. Obviously, the user cannot detect the dishonest behavior of the transformation server. However, the authors did not provide a formal framework for analyzing verifiability, and their method could only work in a heuristic model, namely random oracle model [15]. In addition, their method would be infeasible in the standard model, since their constructions of outsourced ABE in the standard model did not support recovery of the encryption randomness. Recently, Lai et al. [9] formalized a security model for capturing the modification in an outsourced ABE system and proposed a concrete construction with verifiable outsourced decryption. Their construction appends a redundant ciphertext of a random message and a tag (computed from the real message and the random message) to each ciphertext, and requires the original untransformed ciphertext as an auxiliary input in the final decryption step by the user. Compared with Green et al.'s outsourced ABE scheme [8], the scheme in Lai et al. [9] introduces significant overhead in both ciphertext size and decryption operation (see the comparison in Table I). In [16], the authors gave an efficient method to check the correctness of the outsourced decryption in a distributed system. It requires more than one key generation service provider (KGSP) and at least one KGSP honestly takes the right ciphertext as input. Otherwise, their verification model suffers from the attack as existed in Green et al.'s security model.

A. Main Contribution

In this paper, we present an efficient method to verify the correctness of the transformed ciphertext in an ABE system with outsourced decryption. Specifically, our approach is built on an outsourced ABE scheme that works in the key encapsulated mechanism (KEM) setting where the ABE ciphertext encrypts a symmetric session key. Instead of using the encapsulated session key to symmetrically encrypt a message, we first compress it to a shorter string using a hash function, and the output of the hash function is later used to check the correctness of the session key. However, the hash value makes the session key loss some entropy and makes it no longer uniform. Hence, we apply a key extractor to the non-uniform session key to extract a nearly uniform symmetric key, called symmetric encryption key, which is used to encrypt the message. To verify the integrity of the symmetric encrypted ciphertext, we compute a hash value on the concatenation of the ciphertext and the hash value mentioned above. The second hash value is then used to verify the correctness of the outsourced decryption. This method works well as long as the hash function is collision-resistant. Intuitively, the second hash value guarantees the integrity of the symmetric encrypted ciphertext and the first hash value implicitly guarantees that the symmetric session key is correct, which in turn guarantees the correctness of the symmetric encryption key and hence the recovered message.

We provide formal proofs of the (selective) chosen-plaintext security and the verifiability in the standard model, which is a slight modification of the security model first proposed in [9] for verifiable outsourced ABE. In our security model, we explicitly specify the encryption algorithm to output a ciphertext together with a verification key. The verification key resists modification and serves as an auxiliary input of the decryption algorithm to check the correctness of the outsourced computation. In our construction, the verification key is just the second hash value mentioned above. We stress that any ABE system without such an auxiliary key as input can not check the correctness of the outsourced computation, since any one can generate a valid ciphertext of a different message to replace the ciphertext the user intends to decrypt. Indeed, previous constructions implicitly view the original ciphertext (or part of it) as the verification key.

Our approach applies to the construction of both verifiable outsourced CP-ABE and verifiable outsourced KP-ABE. To keep the paper compact, we present a concrete construction of CP-ABE scheme with verifiable outsourced decryption based on Green et al.'s CP-ABE with outsourced decryption but without verifiability [8]. To validate the advantage of our approach, we implement our scheme on an Intel Core PC environment to test the performances of both the local client and the proxy server (i.e., the cloud transformation server). The performance results as given in Table II indicates that our approach is almost optimal in converting a non-verifiable outsourced ABE scheme to a verifiable one.

B. Organization

The rest of this paper is organized as follows. In Section II, we recall the basic definitions of cryptographic primitives used in this paper. The definition and security model for ABE scheme is given in Section III. The outsourced ABE scheme with efficient verification is proposed in Section IV. We demonstrate the efficiency of our scheme in Section V. Finally, we conclude this paper in Section VI.

II. Background

A. Access Structures

Definition 1 (Access Structure [17]): Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in A$. An access structure (respectively, monotone access structure) is a collection (resp. monotone collection) $\mathbb{A}$ of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1,P_2,\ldots,P_n\}} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets.

In our context, the role of the parties is taken by attributes. Thus, the access structure $\mathbb{A}$ will contain the authorized sets of attributes. We restrict our attention to monotone access structures. However, it is also possible to (inefficiently) realize general access structures using our techniques by having the NOT of an attribute as a separate attribute. Thus, the number of attributes in the system will be doubled. From now on, unless stated otherwise, by an access structure we mean a monotone access structure.

B. Symmetric Encryption

A symmetric encryption (SE) scheme with key space $\mathcal{K}$ consists of two probabilistic polynomial time (PPT) algorithms: $\mathsf{SE.Enc}(K, m)$, mapping a key $K \in \mathcal{K}$ and a message $m \in \{0,1\}^*$ to a ciphertext C, and $\mathsf{SE.Dec}(K, c)$ recovering m from C using K.

We say that a one-time symmetric encryption scheme is semantically secure, if for any PPT adversary $\mathcal{A}$

$$\mathsf{Adv}^{\mathrm{ss}}_{\mathsf{SE},\mathcal{A}}(\kappa) := \left| \Pr\left[b = b' : \begin{array}{c} (m_0, m_1) \leftarrow \mathcal{A}(1^\kappa) \\ b \leftarrow_R \{0,1\}, K \leftarrow_R \mathcal{K} \\ c \leftarrow \mathsf{SE.Enc}(K, m_b) \\ b' \leftarrow \mathcal{A}(c) \end{array} \right] - \frac{1}{2} \right|$$

is negligible in κ.

A semantically secure one-time SE can be simply constructed from any pseudorandom generator using the one-time pad encryption scheme.

C. Randomness Extractor

For a discrete distribution X over Ω, we denote its min-entropy by $\mathrm{H}_\infty(X) = -\log(\max_{\omega\in\Omega} \Pr[X = \omega])$. The *average min-entropy* of X conditioned on Y is defined as $\widetilde{\mathrm{H}}_\infty(X|Y) = -\log(E_{y\leftarrow Y}[2^{-\mathrm{H}_\infty(X|Y=y)}])$.

We recall a useful lemma that will be used in our proof.

Lemma 1 [18]: Let X, Y and Z be random variables. If Y has at most 2^r possible values, then $\widetilde{\mathrm{H}}_\infty(X|(Y,Z)) \geq \widetilde{\mathrm{H}}_\infty(X|Z) - r$.

1) Randomness Extractor: An efficient function $\mathsf{Ext} : \mathcal{X} \times \{0,1\}^{\bar{t}} \to \mathcal{Y}$ is an average-case (k, ϵ)-strong extractor if for all random variables (X, Z) such that $X \in \mathcal{X}$ and $\widetilde{\mathrm{H}}_\infty(X|Z) \geq k$, we have $SD((Z, s, \mathsf{Ext}(X, s)), (Z, s, U_\mathcal{Y})) \leq \epsilon$, where $s \leftarrow_R \{0,1\}^{\bar{t}}$, $U_\mathcal{Y} \leftarrow_R \mathcal{Y}$, and $SD(\cdot,\cdot)$ denotes the statistical distance between two distributions.

By the leftover hash lemma [18], any family of pairwise independent hash functions $\mathcal{H} := \{h : \mathcal{X} \to \mathcal{Y}\}$ is an average-case $(\widetilde{\mathrm{H}}_\infty(X|Z), \epsilon)$-strong extractor if $\widetilde{\mathrm{H}}_\infty(X|Z) \geq \log|\mathcal{Y}| + 2\log(1/\epsilon)$.

III. New Definition of Verifiable Outsourced Decryption for ABE

Let $\mathbb{A}$ denote an access structure and S denote a set of attributes. For generality, we will define I_{enc} and I_{key} as the inputs to the encryption algorithm and the key generation algorithm, respectively. Concretely, we have

$$(I_{enc}, I_{key}) := \begin{cases} (\mathbb{A}, S) & \text{in a CP-ABE setting} \\ (S, \mathbb{A}) & \text{in a KP-ABE setting} \end{cases}$$

We also denote by $f(I_{key}, I_{enc}) = 1$ the case $S \in \mathbb{A}$ and by $f(I_{key}, I_{enc}) = 0$ otherwise.

Definition 2 (Syntax of VO-ABE): An ABE system with *privately* verifiable outsourced decryption (shorted as VO-ABE system) consists of the following PPT algorithms:

- $(\mathrm{MPK}, \mathrm{MSK}) \leftarrow \mathsf{Setup}(\kappa, U)$: The setup algorithm takes as input a security parameter κ and an attribute universe description U. It outputs a master public key MPK (which defines a message space $\mathcal{M}$) and a master secret key MSK.
- $(\mathrm{CT}_{I_{enc}}, \mathrm{VK_M}) \leftarrow \mathsf{Encrypt}(\mathrm{MPK}, \mathrm{M}, I_{enc})$: The encryption algorithm takes as input MPK, $\mathrm{M} \in \mathcal{M}$ and I_{enc}. It outputs a ciphertext $\mathrm{CT}_{I_{enc}}$ and a verification key $\mathrm{VK_M}$.
- $(\mathrm{TK}_{I_{key}}, \mathrm{DK}_{I_{key}}) \leftarrow \mathsf{KeyGen}(\mathrm{MSK}, I_{key})$: The transformation key generation algorithm takes as input

MSK and I_{key}. It outputs a transformation key $\text{TK}_{I_{key}}$ and a decryption key $\text{DK}_{I_{key}}$.

- $\text{CT}_{out} \leftarrow \mathsf{Transform}(\text{TK}_{I_{key}}, \text{CT}_{I_{enc}})$: The ciphertext transformation algorithm takes as input $\text{TK}_{I_{key}}$ and $\text{CT}_{I_{enc}}$. It outputs a partially decrypted ciphertext CT_{out}.
- $\text{M}/\perp \leftarrow \mathsf{Decrypt}(\text{DK}_{I_{key}}, \text{VK}_{\text{M}}, \text{CT}_{out})$: The message recovering algorithm takes as input VK_{M}, $\text{CT}^{out}_{I_{enc}}$ and $\text{DK}_{I_{key}}$. It outputs a message $\text{M} \in \mathcal{M} \cup \{\perp\}$. Here the special symbol $\perp$ indicates that the partially decrypted ciphertext is invalid.

Remark 1: We refer to the above outsourced ABE as *privately verifiable* ABE, since the verification (in the decryption algorithm) takes care of the decryption key. We similarly define a *publicly verifiable* outsourced ABE system if there is a verification algorithm that does not take any secret information as input. Observe that any outsourced ABE without taking a verification key as an auxiliary input in the decryption algorithm cannot provide a guarantee on correctness of the transformation. The reason is that, in the public key setting, anyone can encrypt any message into a valid ciphertext and compute the corresponding verification key, and then use them to replace the designated ciphertext and verification key. An interesting problem is to minimize the size of the verification key to save the storage and transmission bandwidth of a client device.

A. Outsourced ABE (in the KEM Setting)

If an outsourced ABE with verification does not have a verification key (i.e., $\text{VK} = \emptyset$), we refer to it as the standard notion of outsourced ABE (without verification) [8]. We denote by $\mathsf{ABE}_O = (\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{KeyGen}, \mathsf{Transform}, \mathsf{Decrypt})$ an outsourced ABE system. In Section IV, the outsourced ABE essentially works in the KEM setting, where the ABE ciphertext hides a symmetric session key. The formal definition of attribute-based KEM with outsourced decryption is exactly the same as that of ABE with outsourced decryption, except that the encryption algorithm of ABE is replaced by an encapsulation algorithm, which doesn't take a message as an input. We show that any outsourced ABE system can be simply converted to an attribute-based KEM with outsourced decryption via the following method: the encapsulation algorithm takes as input MPK and I_{enc}. It first chooses a random session key (message) K from $\mathcal{M}$, and then computes $\text{CT}_{I_{enc}} \leftarrow \mathsf{Encrypt}(\text{MPK}, K, I_{enc})$ using the encryption algorithm of the outsourced ABE. Finally, it outputs $(\text{CT}_{I_{enc}}, K)$.

B. Correctness

For a VO-ABE system, the correctness requires that for any message $\text{M} \in \mathcal{M}$, if all parties follow the prescribed algorithms in Definition 2, the final decryption algorithm should output the same message M.

C. RCCA Security Experiment

We adapt the replayable chosen-ciphertext attack (RCCA) security [8] for ABE with outsourced decryption to the one for ABE with verifiable outsourced decryption. The difference between them is that the adversary also obtains an additional information, i.e., a verification key, which may be related to the message encrypted in the challenge ciphertext.

Setup. The challenger runs the $\mathsf{Setup}(\kappa, U)$ to generate the master key pair (MPK, MSK). It gives MPK to the adversary.

Phase 1. The challenger initializes an empty table T, an empty set D and a counter $j := 0$. Proceeding adaptively, the adversary can repeatedly make any of the following queries:

- $\mathsf{Create}(I_{key})$: The challenger sets $j := j+1$. It first runs $\mathsf{KeyGen}(\text{MSK}, I_{key})$ to obtain a transformation key pair $(\text{TK}_{I_{key}}, \text{DK}_{I_{key}})$. Then, it returns $\text{TK}_{I_{key}}$ to the adversary and stores in table T the entry $(j, I_{key}, \text{TK}_{I_{key}}, \text{DK}_{I_{key}})$.
- $\mathsf{Corrupt}(i)$: If there exists an i^{th} entry in table T, then the challenger obtains the entry $(i, I_{key}, \text{TK}_{I_{key}}, \text{DK}_{I_{key}})$. If $f(I^*_{enc}, I_{key}) = 1$, it returns $\perp$; otherwise it returns the decryption key $\text{DK}_{I_{key}}$ to the adversary and sets $D := D \cup \{I_{key}\}$. If no such entry exists, then it returns $\perp$.
- $\mathsf{Decrypt}(i, (\text{VK}_{\text{M}}, \text{CT}_{out}))$: If there exists an i^{th} entry in table T, then the challenger obtains the entry $(i, I_{key}, \text{TK}_{I_{key}}, \text{DK}_{I_{key}})$ and returns the output of $\mathsf{Decrypt}(\text{DK}_{I_{key}}, \text{VK}_{\text{M}}, \text{CT}_{out})$ to the adversary. If no such entry exists, then it returns $\perp$.

Challenge. The adversary submits two equal-length messages M_0 and M_1 as well as a challenge I^*_{enc} such that for all $I_{key} \in D$, $f(I^*_{enc}, I_{key}) \neq 1$. The challenger picks a bit $b \in \{0,1\}$ uniformly at random, and then runs $\mathsf{Encrypt}(\text{MPK}, \text{M}_b, I^*_{enc})$ to obtain a challenge ciphertext $\text{CT}^*_{I^*_{enc}}$ and verification key $\text{VK}^*_{\text{M}_b}$ of message M_b under I^*_{enc}. It returns $(\text{CT}^*_{I^*_{enc}}, \text{VK}^*_{\text{M}_b})$ to the adversary.

Phase 2. The same as Phase 1 except that:

- the adversary cannot issue a Corrupt query that would result in a value I_{key} such that $f(I^*_{enc}, I_{key}) = 1$ and $I_{key} \in D$.
- if a decryption response would be either M_0 or M_1, then the challenger responds with the special message $\perp$.

Guess. The adversary outputs a guess b' of b.

Fig. 2. RCCA-security game for ABE with verifiable outsourced decryption.

We say that $\mathcal{A}$ succeeds in the game defined in Fig. 2 if $b = b'$. The advantage of $\mathcal{A}$ is defined as

$$\mathsf{Adv}^{\text{rcca}}_{\mathsf{ABE}_{out},\mathcal{A}}(\kappa) := \left|\Pr[b = b'] - \frac{1}{2}\right|.$$

Definition 3 (RCCA-Security): An ABE system with verifiable outsourced decryption is RCCA-secure, if for all PPT adversary $\mathcal{A}$, the function $\mathsf{Adv}^{\text{rcca}}_{\mathsf{ABE}_{out},\mathcal{A}}(\kappa)$ is negligible in κ.

D. CPA Security

We say that an ABE scheme with verifiable outsourced decryption is CPA-secure (or semantically secure against chosen-plaintext attacks) if we remove the decryption queries in both Phase 1 and Phase 2.

Setup. The challenger runs $\mathsf{Setup}(\kappa, U)$ to generate the master key pair (MPK, MSK). It sends MPK to the adversary.
Phase 1. The challenger can adaptively query the Create, Corrupt and Decrypt oracles as in **Phase 1** in Fig. 2.
Challenge. The adversary submits a message M^* and an value I^*_{enc}. The challenger computes a challenge ciphertext $(\text{CT}^*_{I^*_{enc}}, \text{VK}^*_{\text{M}^*}) = \mathsf{Encrypt}(\text{MPK}, \text{M}^*, I^*_{enc})$ and sends it to $\mathcal{A}$.
Phase 2. The same as Phase 1
Output. Finally, the adversary outputs a value I^*_{key} such that $f(I^*_{enc}, I^*_{key}) = 1$ and a transformation ciphertext CT_{out}.

Fig. 3. Verifiability game for ABE with outsourced decryption.

E. Selective CPA Security

We say that an ABE scheme with verifiable outsourced decryption is selective CPA-secure if we add an **Initiation** stage before the **Setup** where the adversary commits to the challenge value I^*_{enc}.

If $\text{VK} = \emptyset$, the above security notions for ABE with verifiable outsourced decryption are exactly those for the standard ABE with outsourced decryption.

F. Verifiability

Fig. 3 presents a formal definition of the verifiability for an ABE system with outsourced decryption, through a game played between an adversary $\mathcal{A}$ and a challenger.

Suppose that the entry $(j^*, I^*_{key}, \text{TK}_{I^*_{key}}, \text{DK}_{I^*_{key}})$ has already existed in table T. If not, the challenger can generate it using MSK. We say that $\mathcal{A}$ succeeds in the game defined in Fig. 3, if $\mathsf{Decrypt}(\text{DK}_{I^*_{key}}, \text{VK}^*_{\text{M}^*}, \text{CT}_{out}) \notin \{\text{M}^*, \perp\}$. $\mathcal{A}$'s advantage is defined as

$$\mathsf{Adv}^{\mathsf{vrfy}}_{\mathsf{ABE}_{out},\mathcal{A}}(\kappa) := \Pr\big[\mathcal{A} \text{ succeeds in Fig. 3}\big].$$

Definition 4 (Verifiability): An ABE system with outsourced decryption is (privately) verifiable, if for all PPT adversary $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathsf{vrfy}}_{\mathsf{ABE}_{out},\mathcal{A}}(\kappa)$ is negligible in κ.

G. Weak Verifiability

The above definition of verifiability allows the adversary to obtain a decryption key of the challenge ciphertext. This may be a bit too strong, as in practice, the proxy only knows the transformation key. So, we can restrict the adversary's ability the same as that of the adversary in the model of RCCA security, i.e., the adversary is forbidden to query the Corrupt oracle on any value I_{key} such that $f(I^*_{enc}, I_{key}) = 1$. We can further weaken the adversary's ability to the selective setting, i.e., the adversary must commit the challenge value I^*_{enc} at the beginning of the game. Consequently, it would be possible to design an outsourced ABE with short verification key. Nevertheless, in this paper, we consider the strongest verifiability model, which encompasses more powerful adversary, such as a valid user in the ABE system.

IV. Verifiable Outsourced Decryption

In this section, we first present a generic method to construct ABE schemes with verifiable outsourced decryption. Then, we provide a formal security proofs of its (selective) CPA-security and verifiability. Finally, we present an instantiation based on the outsourced CP-ABE scheme proposed by Green et al. [8].

Our generic construction uses the following components.

- An outsourced ABE system: ABE_O=(Setup', $\mathsf{Encrypt}'$, KeyGen', $\mathsf{Transform}'$, $\mathsf{Decrypt}'$), with message space $\mathcal{M}$.
- Two collision-resistant hash functions:
$$\mathsf{H}_0 : \mathcal{M} \to \{0,1\}^{\ell_{\mathsf{H}_0}}, \quad \mathsf{H} : \{0,1\}^* \to \{0,1\}^{\ell_{\mathsf{H}}}.$$
- A symmetric encryption (SE) scheme SE=($\mathsf{SE.Enc}$, $\mathsf{SE.Dec}$) with key space $\{0,1\}^{\ell_{\mathsf{SE}}}$.
- A family of pairwise independent hash functions $\mathcal{H}$ from $\mathcal{M}$ to $\{0,1\}^{\ell_{\mathsf{SE}}}$.

The above parameters satisfy the following condition:

$$0 < \ell_{\mathsf{SE}} \leq (\log|\mathcal{K}| - \ell_{\mathsf{H}_0}) - 2\log(1/\epsilon_{\mathcal{H}})$$

where $\epsilon_{\mathcal{H}}$ is a negligible value in κ.

A. Generic Construction

We first depict the framework of our approach in Fig. 4. In framework, the underlying outsourced ABE system works in the KEM setting, i.e., it encrypts a random key $K \in \mathcal{M}$ rather than a real message. Instead of using the random key to symmetrically encrypt data, we first compute an *intermediate* verification key $\mathsf{H}_0(K)$, and then use an extractor h to derive a symmetric key $K_{\mathsf{SE}} = h(K)$. Finally, we use a symmetric encryption scheme to hide the real message, i.e., $C_{\mathsf{SE}} = \mathsf{SE.Enc}(K_{\mathsf{SE}}, M)$ and compute the verification key $\text{VK} = \mathsf{H}(\mathsf{H}_0(K)||C_{\mathsf{SE}})$. By the collision resistance of H and H_0, if VK is untampered, so are C_{SE} and $\mathsf{H}_0(K)$. The later implicitly guarantees the correctness of the recovered random key K. By the semantic security of the underlying ABE system, K is computationally indistinguishable from a truly random key $R \in \mathcal{M}$. However, this statement fails as the verification key may reveal information on K. Recall that VK leaks at most ℓ_{H_0}-bit information of K. So, applying an average-case extractor h, we may derive a new random key, i.e., the symmetric key K_{SE}.

Next, we formally describe our verifiable outsourced ABE system ABE_{VO}=(Setup, $\mathsf{Encrypt}$, KeyGen, $\mathsf{Transform}$, $\mathsf{Decrypt}$) in Fig. 5 and its security proof.

1) Correctness: The correctness of the above ABE system follows directly from the correctness of the underlying outsourced ABE system ABE_O and the correctness of the underlying symmetric encryption scheme SE.

2) Efficiency: Compared with the underlying ABE system, the new ABE only introduces one hash value (i.e., the verification key) to the ciphertext and two hash value computations in the final decryption operation at the user side.

B. Security Analysis

Theorem 1: Suppose that the underlying outsourced ABE system is (selectively) CPA-secure, $\mathcal{H}$ is a family of pairwise independent hash functions, SE *is a semantically secure one-time symmetric encryption scheme, the parameters satisfy*

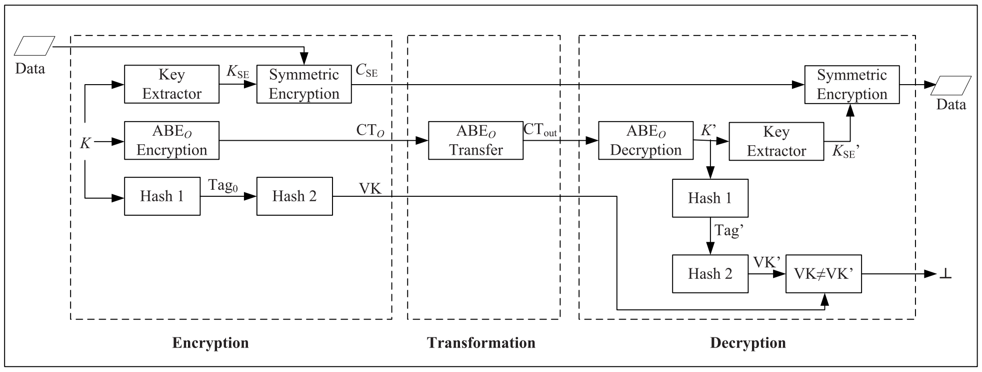


Fig. 4. The framework of generic construction of verifiable outsourced ABE.

- $(\text{MPK}, \text{MSK}) \leftarrow \mathsf{Setup}(1^\kappa, U)$. Run $\mathsf{Setup}'(1^\kappa, U)$ to obtain a master key pair $(\text{MPK}', \text{MSK}')$ of the outsourced ABE system. Then, choose an extractor $h \in \mathcal{H}$, two hash functions H_0 and H, and a symmetric encryption scheme SE. Finally, output $\text{MPK} = (\text{MPK}', \mathsf{H}_0, \mathsf{H}, h, \mathsf{SE})$ and $\text{MSK} = (\text{MPK}, \text{MSK}')$.
- $(\text{CT}_{I_{enc}}, \text{VK}_\text{M}) \leftarrow \mathsf{Encrypt}(\text{MPK}, \text{M}, I_{enc})$. Choose a random key $K \in \mathcal{M}$ and then run $\mathsf{Encrypt}'(\text{MPK}', K, I_{enc})$ to obtain a ciphertext $\text{CT}'_{I_{enc}}$. Next, compute $\mathsf{Tag}_0 = \mathsf{H}_0(K')$, $K_{\mathsf{SE}} = h(K')$, $C_{\mathsf{SE}} = \mathsf{SE.Enc}(K_{\mathsf{SE}}, \text{M})$, and $\mathsf{Tag} = \mathsf{H}(\mathsf{Tag}_0 || C_{\mathsf{SE}})$. Finally, output $\text{CT}_{I_{enc}} = (\text{CT}'_{I_{enc}}, C_{\mathsf{SE}})$ and $\text{VK}_\text{M} = \mathsf{Tag}$.
- $(\text{TK}_{I_{key}}, \text{DK}_{I_{key}}) \leftarrow \mathsf{KeyGen}(\text{MSK}, I_{key})$. The transformation key generation algorithm is identical to that of the underlying outsource ABE ciphertext, i.e.,
$$(\text{TK}_{I_{key}}, \text{DK}_{I_{key}}) \leftarrow \mathsf{KeyGen}'(\text{MSK}', I_{key}).$$
- $\text{CT}_{out} \leftarrow \mathsf{Transform}(\text{TK}_{I_{key}}, \text{CT}_{I_{enc}})$. First, compute a partial decryption CT'_{out} via $\mathsf{Transform}'(\text{TK}_{I_{key}}, \text{CT}'_{I_{enc}})$. Return $\text{CT}_{out} = (\text{CT}'_{out}, C_{\mathsf{SE}})$.
- $\text{M}/\perp \leftarrow \mathsf{Decrypt}(\text{DK}_{I_{key}}, \text{VK}_\text{M}, \text{CT}_{out})$. First, recover a random key K from $\mathsf{Decrypt}'(\text{DK}_{I_{key}}, \text{CT}'_{out})$. Then, compute $\mathsf{Tag}_0 = \mathsf{H}_0(K)$. If $\mathsf{H}(\mathsf{Tag}_0 || C_{\mathsf{SE}}) \neq \text{VK}$, return $\perp$. Otherwise, compute $K_{\mathsf{SE}} = h(K)$ and return $\text{M} = \mathsf{SE.Dec}(K_{\mathsf{SE}}, C_{\mathsf{SE}})$.

Fig. 5. Generic construction of verifiable outsourced ABE.

$0 < \ell_{\mathsf{SE}} \leq (\log|\mathcal{M}| - \ell_{\mathsf{H}_0}) - 2\log(1/\epsilon_{\mathcal{H}})$. *Then, the above ABE system with outsourced decryption is (selectively) CPA-secure.*

Proof: The proof applies the hybrid argument of games [19]. We begin by defining three games: Game_0, Game_1 and Game_2. Game_0 is the original (selective) CPA-security game as defined in Fig. 2 for an outsourced ABE system. We aim to prove that there is only a negligible difference between any two consecutive games from the adversary's point of views, and in Game_2 the adversary has a negligible probability to distinguish the encryption of message M_0 from encryption of message M_1. Let S_i denote $\mathcal{A}$'s success probability in Game_i.

- Game_0: This is the original (selective) CPA-security game. Hereafter, let $(\text{CT}^*_{I^*_{enc}}, \text{VK}^*_\text{M}) = ((\text{CT}'^*_{I^*_{enc}}, C^*_{\mathsf{SE}}), \mathsf{Tag}^*)$ denotes the challenge ciphertext and verification key for a challenge value I^*_{enc} selected by the adversary. We also denote by $K^* \in \mathcal{M}$ the key encrypted in ciphertext $\text{CT}'^*_{I^*_{enc}}$ and by $K^*_{\mathsf{SE}} = h^*(K^*)$ the symmetric key used in ciphertext C^*_{SE}.
- Game_1: This game is the same as Game_0, except that we compute Tag^*_0 and K^*_{SE} using another random key $R^* \in \mathcal{M}$ which is independent of K^* encrypted in $\text{CT}'^*_{I^*_{enc}}$.
- Game_2: This game is the same as Game_1, except that K^*_{SE} is replaced by a truly random string $R^*_{\mathsf{SE}} \in \{0,1\}^{\ell_{\mathsf{SE}}}$.

Claim 1: Suppose that the outsourced ABE system is (selectively) CPA-secure, then the adversary's views in Game_0 *and* Game_1 *are computationally indistinguishable.*

Proof of Claim 1: We define a PPT algorithm Sim which aims to break the (selective) CPA-security of the underlying ABE system under the help of the adversary $\mathcal{A}$. Sim simulates $\mathcal{A}$'s views in Game_0 or in Game_1 depending on its challenge ciphertext. Denote by Chall_O the challenger of the underlying ABE system.

- **Setup.** Sim first runs Chall_O to obtain a challenge public parameter MPK'^*. Then, it chooses by itself two collision-resistant hash functions H^*_0 and H^*, a random extractor $h^* \in \mathcal{H}$ and a semantically secure one-time encryption scheme SE^*. Finally, it sends $(\text{MPK}'^*, \mathsf{H}^*_0, \mathsf{H}^*, h^*, \mathsf{SE}^*)$ to $\mathcal{A}$ as a challenge master public key.
- **Phase 1.** It is straightforward to answer $\mathcal{A}$'s queries, including $\mathsf{Create}(I_{key})$ (for any value I_{key}) and $\mathsf{Corrupt}(i)$. This is because Sim can obtain the answers of these queries via running Chall_O with the same queries.
- **Challenge.** Once $\mathcal{A}$ submits two equal-length messages M_0 and M_1 as well as a value I^*_{enc}, the simulator Sim first chooses two independent random keys K^*, $R^* \in \mathcal{M}$. It then queries Chall_O with $((K^*, R^*), I^*_{enc})$. Chall_O will return a challenge ciphertext $\text{CT}'^*_{I^*_{enc}}$ to the simulator. Next, Sim sets $K^*_{\mathsf{SE}} = h^*(R^*)$, and $\mathsf{Tag}^*_0 = \mathsf{H}^*_0(R^*)$. It also computes $C^*_{\mathsf{SE}} = \mathsf{SE}^*(K^*_{\mathsf{SE}}, \text{M}_b)$ for a random $b \in \{0,1\}$ and sets $\text{VK}^*_\text{M} = \mathsf{H}^*(\mathsf{Tag}^*_0 || C^*_{\mathsf{SE}})$. Finally, it sends $\text{CT}^*_{I^*_{enc}} = (\text{CT}'^*_{I^*_{enc}}, C^*_{\mathsf{SE}})$ and VK^*_M to the adversary. Clearly, if the "message" encrypted in $\text{CT}'^*_{I^*_{enc}}$ is R^*, then $\text{CT}^*_{I^*_{enc}}$ is a challenge ciphertext

as in Game_0. Otherwise, it is a challenge ciphertext as in Game_1.

- **Phase 2.** The same as Phase 1, except that $\mathcal{A}$ can not query $\mathsf{Corrupt}(i)$, in which the value I_{key} satisfies $f(I^*_{enc}, I_{key}) = 1$.

Finally, Sim outputs what $\mathcal{A}$ outputs. From the above analysis, Sim perfectly simulates $\mathcal{A}$'s views in Game_0 or Game_1. So, we have the following result:

$$|\Pr[S_0] - \Pr[S_1]| \le 2\mathsf{Adv}^{\text{cpa}}_{\mathsf{ABE}_O,\mathcal{B}}(\kappa)$$

for some adversary $\mathcal{B}$ attacking on the CPA-security of the underlying outsourced ABE system. □

Claim 2: Suppose that $\mathcal{H}$ is a family of pairwise independent hash functions, then the adversary's views in Game_1 and Game_2 are statistically indistinguishable.

Proof of Claim 2: Observe that in both Game_1 and Game_2, R^* is completely independent of $\text{CT}'^*_{I^*_{enc}}$, h^* and the master public key MPK. In addition, $\mathsf{Tag}^*_0 = \mathsf{H}^*_0(R^*)$ has at most $2^{\ell_{\mathsf{H}_0}}$ possible values. Hence, by Lemma 1 we have

$$\begin{aligned}\widetilde{\mathrm{H}}_\infty(R^*|(\text{MPK}, \text{CT}'^*_{I^*_{enc}}, h^*, \mathsf{Tag}^*_0)) \\ \ge \widetilde{\mathrm{H}}_\infty(R^*|(\text{MPK}, \text{CT}'^*_{I^*_{enc}}, h^*)) - \ell_{\mathsf{H}_0} = \log|\mathcal{M}| - \ell_{\mathsf{H}_0}.\end{aligned}$$

Since $0 < \ell_{\mathsf{SE}} \le (\log|\mathcal{M}| - \ell_{\mathsf{H}_0}) - 2\log 1/\epsilon_{\mathcal{H}}$, from the adversary's point of view (except the variable C^*_{SE}), the symmetric key K^*_{SE} extracted by $h^*(R^*)$ is $\epsilon_{\mathcal{H}}$-statistically indistinguishable from a truly random symmetric key $R^*_{\mathsf{SE}} \in \{0,1\}^{\ell_{\mathsf{SE}}}$. Observe that the C^*_{SE} is the function of K^*_{SE}, and Tag^* is the function of Tag^*_0 and C^*_{SE}. So, they do not increase the distance between the above two distributions. Therefore, $|\Pr[S_1] - \Pr[S_2]| \le \epsilon_{\mathcal{H}}$. □

Claim 3: Suppose that the symmetric encryption scheme SE is semantically secure, then the adversary in Game_2 has a negligible advantage.

Proof of Claim 3: Observe that in Game_2, the symmetric key R^*_{SE} is truly random. So, we can directly construct an algorithm $\mathcal{B}$ from $\mathcal{A}$ to attack SE^*'s semantic security. Hence, we have $\left|\Pr[S_2] - \frac{1}{2}\right| \le \mathsf{Adv}^{\text{ss}}_{\mathsf{SE}^*,\mathcal{B}}(\kappa)$ for a suitable adversary $\mathcal{B}$ attacking on SE^*'s semantic security. □

Recall that Game_0 is the original (selective) CPA-security game for a verifiable outsourced ABE scheme. So, we have $\mathsf{Adv}^{\text{cpa}}_{\mathsf{ABE}_{VO},\mathcal{A}}(\kappa) := \left|\Pr[S_0] - \frac{1}{2}\right|$.

Taking all claims together, the (selective) CPA-security of the new outsourced ABE follows. □

In the above security proof, we only consider the (selective) CPA-security of our scheme. Similarly, we can proof its RCCA-security if the underlying outsourced ABE scheme is RCCA-secure and the symmetric encryption scheme is also RCCA-secure [20]. However, to date, all RCCA-secure outsourced ABE schemes rely on random oracle (RO) [15]. Hence, the resulting verifiable scheme is just RCCA-secure in the RO model. It may be an independent interest to design an RCCA-secure ABE system with outsourced decryption in the standard model.

Theorem 2: Suppose that H_0 and H are collision-resistant hash functions. Then, the new ABE scheme with outsourced decryption is privately verifiable.

Proof: Given an adversary $\mathcal{A}$ against the verifiability, we construct an efficient algorithm Sim to break the collision-resistance of the underlying hash functions H_0 or H. Given two challenge hash functions $(\mathsf{H}^*_0, \mathsf{H}^*)$, Sim simulates the experiment described in Fig. 3 as follows.

Sim generates the public parameter MPK and master secret key MSK as Setup, except for hash functions H^*_0 and H^*. Note that, Sim knows the master secret key MSK. Hence, it can simulate $\mathcal{A}$'s queries in Phase 1 and Phase 2. For a challenge message M^* and a value I^*_{enc} submitted by $\mathcal{A}$, the simulator first invokes $\mathsf{Encrypt}'(\text{MPK}', K^*, I^*_{enc})$ to obtain a ciphertext $\text{CT}'^*_{I^*_{enc}}$ of a random key $K^* \in \mathcal{M}$. It then sets $\mathsf{Tag}^*_0 = \mathsf{H}^*_0(K^*)$ and $K^*_{\mathsf{SE}} = h^*(K^*)$. Sim also computes $C^*_{\mathsf{SE}} = \mathsf{SKE.Enc}(K^*_{\mathsf{SE}}, \text{M}^*)$ and $\mathsf{Tag}^* = \mathsf{H}^*(\mathsf{Tag}^*_0||C^*_{\mathsf{SE}})$. After that, it sends $\text{CT}^*_{I^*_{enc}} = (\text{CT}'^*_{I^*_{enc}}, C^*_{\mathsf{SE}})$ and $\text{VK}^*_{\text{M}} = \mathsf{Tag}^*$ to the adversary. It holds VK^*_{M} and (K^*, C^*_{SE}). Finally, the adversary outputs a vale I^*_{key} (such that $f(I^*_{enc}, I^*_{key}) = 1$) and a transformation ciphertext $\text{CT}_{out} = (\text{CT}'_{out}, C_{\mathsf{SE}})$. If $\mathcal{A}$ breaks the verifiability, Sim will recover a message $\text{M} \notin \{\text{M}^*, \perp\}$ via $\mathsf{Decrypt}(\text{DK}_{I^*_{key}}, \text{VK}^*_{\text{M}}, \text{CT}_{out})$. We now discuss $\mathcal{A}$'s success probability. Observe that the decryption algorithm outputs $\perp$ if $\mathsf{H}^*(\mathsf{Tag}_0||C_{\mathsf{SE}}) \ne \mathsf{Tag}^*$, where $\mathsf{Tag}_0 = \mathsf{H}^*_0(K)$ and $K = \mathsf{Decrypt}'(\text{DK}_{I^*_{key}}, \text{CT}'_{out})$. So, we only need to consider the following two cases:

- Case 1: $(\mathsf{Tag}_0, C_{\mathsf{SE}}) \ne (\mathsf{Tag}^*_0, C^*_{\mathsf{SE}})$. Since Sim knows $(\mathsf{Tag}^*_0, C^*_{\mathsf{SE}})$, if this case occurs, Sim immediately obtains a collision of the hash function H^*.
- Case 2: $(\mathsf{Tag}_0, C_{\mathsf{SE}}) = (\mathsf{Tag}^*_0, C^*_{\mathsf{SE}})$, but $K \ne K^*$. Observe that $\mathsf{H}^*_0(K) = \mathsf{Tag}_0 = \mathsf{Tag}^*_0 = \mathsf{H}^*_0(K^*)$. So, it breaks the collision-resistance of H^*_0.

This completes the proof of Theorem 2. □

C. Instantiation

In this subsection, we present an instantiation of our generic construction based on the outsourced ABE system proposed in [8], which is in turn based on Waters CP-ABE scheme [4]. We begin by introducing some basic notations used in the instantiation.

1) Bilinear Maps: Let $\mathcal{BP}(1^\kappa)$ be a probabilistic polynomial time (PPT) algorithm that takes as input a security parameter 1^κ and outputs a description $(p, \mathbb{G}, g, \mathbb{G}_T, \hat{e})$, where $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative groups of prime order p, g is a generator of $\mathbb{G}$ and $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map with the properties:

1) Bilinearity: for all $g_1, g_2 \in \mathbb{G}$ and all $a, b \in \mathbb{Z}_p$, we have $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$.
2) Non-degeneracy: $\hat{e}(g, g) \ne 1$.
3) Computability: Group operations in $\mathbb{G}$ and $\mathbb{G}_T$, and the bilinear map $\hat{e}$ are efficiently computable.

We refer to $\mathbb{G}$ as a bilinear group. Note that, the map $\hat{e}$ is symmetric since $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1^b, g_2^a)$.

2) Linear Secret Sharing Schemes: We will make use of linear secret-sharing schemes. We adopt the definition from [17].

Definition 5 (Linear Secret-Sharing Schemes (LSSS)): A secret-sharing scheme Π over a set of parties $\mathcal{P}$ is called linear (over $\mathbb{Z}_p$) if

- $(\text{MPK}, \text{MSK}) \leftarrow \mathsf{Setup}(1^\kappa, U)$: It runs $\mathsf{Setup}'(1^\kappa, U)$ to obtain $(\text{MPK}', \text{MSK}')$. Specifically, Setup' first runs $\mathcal{BP}(1^\kappa)$ to obtain a description of bilinear map $(p, \mathbb{G}, g, \mathbb{G}_T, \hat{e})$, and then chooses U random group elements $h_1, \ldots, h_U \in \mathbb{G}$. It also chooses random exponents $\alpha, a \in \mathbb{Z}_p$ and sets $\text{MPK}' = (g, \hat{e}(g,g)^\alpha, g^a, h_1, \ldots, h_U)$ and $\text{MSK}' = (\text{MPK}', g^\alpha)$. Setup outputs a public parameters $\text{MPK} = (\text{MPK}', \mathsf{H}_0, \mathsf{H}, h, \mathsf{SE})$ and a master secret key $\text{MSK} = (\text{MPK}, g^\alpha)$.
- $(\text{CT}_{\mathbb{A}}, \text{VK}_{\text{M}}) \leftarrow \mathsf{Encrypt}(\text{MPK}, \text{M}, (\mathbf{A}, \rho))$. It runs $\mathsf{Encrypt}'(\text{MPK}', R, (\mathbf{A}, \rho))$ to obtain an ABE ciphertext $\text{CT}'_{\mathbb{A}}$ of a random message $R \in \mathbb{G}_T$. Let $\mathbf{A}$ be an $\ell \times n$ matrix. The algorithm $\mathsf{Encrypt}'$ works as follows: it first picks a random column vector $\vec{v} = (s, y_2, \ldots, y_n) \in \mathbb{Z}_p^n$, and calculates $\lambda_i = A_i \vec{v}$, where A_i is the vector corresponding to i-th row of the matrix $\mathbf{A}$. It also randomly chooses $r_1, \ldots, r_\ell \in \mathbb{Z}_p$. The ciphertext $\text{CT}'_{\mathbb{A}} = (\mathbb{A}, C, C', (C_1, D_1), \ldots, (C_\ell, D_\ell))$ is computed as follows:
$$\mathbb{A} = (\mathbf{A}, \rho), \quad C = \hat{e}(g,g)^{\alpha s} \cdot R, \quad C' = g^s$$
$$(C_1, D_1) = (g^{a\lambda_1} h_{\rho(1)}^{-r_1}, g^{r_1}), \ldots, (C_\ell, D_\ell) = (g^{a\lambda_\ell} h_{\rho(\ell)}^{-r_\ell}, g^{r_\ell}).$$
After that, Encrypt sets $\mathsf{Tag}_0 = \mathsf{H}_0(R)$, and computes a symmetric key $K_{\mathsf{SE}} = h(R)$. Next, Encrypt computes $C_{\mathsf{SE}} = \mathsf{SEK.Enc}(K_{\mathsf{SE}}, \text{M})$ and $\mathsf{Tag} = \mathsf{H}(\mathsf{Tag}_0 \| C_{\mathsf{SE}})$. Finally, Encrypt outputs the ciphertext $\text{CT}_{\mathbb{A}} = (\text{CT}'_{\mathbb{A}}, C_{\mathsf{SE}})$ as well as a verification key $\text{VK}_{\text{M}} = \mathsf{Tag}$.
- $(\text{TK}_S, \text{DK}_S) \leftarrow \mathsf{KeyGen}(\text{MSK}, S)$: It runs $\mathsf{KeyGen}'(\text{MSK}', S)$ and outputs
$$\text{TK}_S = (K, L, \{K_x\}_{x \in S}) = (g^{\alpha/\alpha'} g^{at}, g^t, \{h_x^t\}_{x \in S}) \qquad \text{DK}_S = \alpha'.$$
where $t, \alpha' \in \mathbb{Z}_p$ are two random exponents.
- $\text{CT}_{out} \leftarrow \mathsf{Transform}(\text{TK}_S, \text{CT}_{\mathbb{A}})$. It first runs $\mathsf{Transform}'(\text{TK}_S, \text{CT}_{\mathbb{A}})$ to obtain (C, C'_T), where C'_T is computed as follows
$$C'_T = \frac{\hat{e}(C', K')}{\Pi_{i \in I}(\hat{e}(C_i, L) \cdot \hat{e}(D_i, K_{\rho(i)}))^{\omega_i}} = \frac{\hat{e}(g,g)^{(\alpha/\alpha')s} \cdot \hat{e}(g,g)^{ast}}{\Pi_{i \in I} \hat{e}(g,g)^{ta\lambda_i\omega_i}} = \hat{e}(g,g)^{(\alpha/\alpha')s}.$$
In above, $I = \{i : \rho(i) \in S\} \subset \{1, \ldots, \ell\}$ and $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ is a set of constants such that $\sum_{i \in I} \omega_i A_i = (1, 0, \ldots, 0)$. Here, we suppose that $f(\mathbb{A}, S) = 1$. Otherwise, the decryption algorithm outputs $\perp$ directly. Finally, Transform outputs $\text{CT}_{out} = (C, C'_T, C_{\mathsf{SE}})$.
- $\text{M}/\perp \leftarrow \mathsf{Decrypt}(\text{DK}_S, \text{VK}_{\text{M}}, \text{CT}_{out})$. It first runs $\mathsf{Decrypt}(\text{DK}_S, (C, C'_T))$ to recovery the random message $R = C/C_T$, where $C_T = {C'_T}^{\alpha'}$. Then it computes $\mathsf{Tag}_0 = \mathsf{H}_0(R)$. If $\mathsf{H}(\mathsf{Tag}_0 \| C_{\mathsf{SE}}) \neq \mathsf{Tag}$, it returns $\perp$ and halts immediately. Otherwise, it computes $K_{\mathsf{SE}} = h(R)$ and returns $\text{M} := \mathsf{SKE.Dec}(K_{\mathsf{SE}}, C_{\mathsf{SE}})$.

Fig. 6. A verifiable outsourced CP-ABE scheme.

1) The shares of the parties form a vector over $\mathbb{Z}_p$.
2) There exists an ℓ rows by n columns matrix $\mathbf{A}$ called the share-generating matrix for Π. There exists a function ρ which maps each row of the matrix to an associated party. That is, for $i = 1, \ldots, \ell$, the value $\rho(i)$ is the party associated with row i. When we consider the column vector $\vec{v} = (s, r_2, \ldots, r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, \ldots, r_n \in \mathbb{Z}_p$ are randomly chosen, then $\mathbf{A}\vec{v}$ is the vector of ℓ shares of the secret s according to Π. The share $(\mathbf{A}\vec{v})_i$ belongs to party $\rho(i)$.

It has been shown in [17] that every linear secret sharing-scheme according to the above definition also enjoys the linear reconstruction property, defined as follows. Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, 2, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, there exist constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if $\{\lambda_i\}_{i \in I}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. In other words, the linear combination $\sum_{i \in I} \omega_i A_i$ results in the "target" vector $(1, 0, \ldots, 0)$, where A_i is the i-th row of the matrix $\mathbf{A}$. It is shown in [17] that these constants $\{\omega_i\}$ can be found in time polynomial in the size of the share-generating matrix $\mathbf{A}$.

However, for any unauthorized sets $S \notin \mathbb{A}$, no such constants $\{\omega_i\}$ exist and the secret s should be information theoretically hidden from the parties in S.

3) The Instantiation: Let $\mathsf{GHW} = (\mathsf{Setup}', \mathsf{Encrypt}', \mathsf{KeyGen}', \mathsf{Transform}', \mathsf{Decrypt}')$ denote the outsourced CP-ABE scheme proposed by Green, Hohenberger and Waters [8, Sec. 3, Fig. 5]. Applying our generic construction, we immediately derive a new CP-ABE scheme supporting both outsourced decryption and verifiability as given in Fig. 6.

In our instantiation, the underlying ABE encrypts one group element, which may not have sufficient (computational) entropy for extracting a symmetric key. Nevertheless, we can simply extend it to a random key with enough entropy via an efficient pseudorandom number generator (e.g., AES).

4) Shrinking the Ciphertext Size: Recall that the underlying outsourced ABE system works in the KEM setting and it encrypts a random message R. So, we can replace the random message R with the internal random element $C_T = \hat{e}(g,g)^{as}$ and remove C from the ciphertext. Such modification cannot affect its semantic security, as $\hat{e}(g,g)^{as}$ is used to protect the privacy of the message R.

V. Performance Comparison

We compare the performance of our scheme with other two outsourced ABE schemes [8], [9] in the setting of key-encapsulation mechanism, which has already been used in these two schemes to reduce the ciphertext size.

TABLE I

EFFICIENCY SUMMARY OF OUTSOURCED CP-ABE SCHEMES

Scheme	Out. CT Size	Out. Dec Ops	Local CT Size	Local Dec Ops	Vrf
GHW11 [8]	$(1+2\ell)\|\mathbb{G}\|$	$\leq (2+\ell)P$+$2\ell E_{\mathbb{G}}$	$\|\mathbb{G}_T\|$	$E_{\mathbb{G}_T}$	✗
LDGW13 [9]	$(3+4\ell)\|\mathbb{G}\|$	$\leq (4+2\ell)P$+$(2+4\ell)E_{\mathbb{G}}$	$2\|\mathbb{G}_T\|+\|\mathbb{G}\|$	$2E_{\mathbb{G}}+2E_{\mathbb{G}_T}$	✓
Ours	$(1+2\ell)\|\mathbb{G}\|+\ell_{\mathsf{H}}$	$\leq (2+\ell)P$+$2\ell E_{\mathbb{G}}$	$\|\mathbb{G}_T\|+\ell_{\mathsf{H}}$	$E_{\mathbb{G}_T}$	✓

TABLE II

PERFORMANCE ON CIPHERTEXT OVERHEAD AND DECRYPTION TIME. (a) OUT. CT OVERHEAD (KBYTES). (b) LOCAL CT OVERHEAD (KBYTES). (c) OUT. DEC OPS (SECONDS). (d) LOCAL DEC OPS (SECONDS)

(a)

Scheme	Vrf	10	20	50	80	100
GHW11 [8]	✗	3.48	6.89	17.13	27.37	34.20
LDGW13 [9]	✓	7.13	13.95	34.43	54.91	68.57
Ours IV-C	✓	3.50	6.91	17.15	27.39	34.22

(b)

10	20	50	80	100
0.19	0.19	0.19	0.19	0.19
0.55	0.55	0.55	0.55	0.55
0.21	0.21	0.21	0.21	0.21

(c)

Scheme	Vrf	10	20	50	80	100
GHW11 [8]	✗	.65	.88	1.59	2.52	3.34
LDGW13 [9]	✓	1.00	1.75	3.16	4.99	6.69
Ours IV-C	✓	.65	.88	1.58	2.52	3.36

(d)

10	20	50	80	100
$< .004$	$< .004$	$< .004$	$< .004$	$< .004$
$< .015$	$< .015$	$< .015$	$< .015$	$< .015$
$< .004$	$< .004$	$< .004$	$< .004$	$< .004$

Specifically, the scheme first encrypts a symmetric key, which is then used to symmetrically encrypt data of arbitrary length. We first give a comparison from the theoretical aspect in Table I.

In the table, ℓ refers to an LSSS access structure with an $\ell \times n$ matrix. ℓ_{H} refers to the output size of a CRH function. For 80-bit security, we can choose $\ell_{\mathsf{H}} = 160$. P, $E_{\mathbb{G}}$, $E_{\mathbb{G}_T}$ denote the maximum amounts of time to compute a pairing, an exponentiation in $\mathbb{G}$ and an exponentiation in $\mathbb{G}_T$, respectively. "Out. CT Size" and "Local CT Size" refer to the sizes of ciphertexts input to the transformation algorithm and the decryption algorithm respectively; While "Out. Dec Ops" and "Local Dec Ops' refer to the computation costs incurred by the transformation algorithm and the decryption algorithm, respectively. We ignore the non-dominant operations. Indeed, we will see shortly (see Table II) that they are negligible in our experimental results.

To validate the advantage of our approach from the practical aspect, we implement our CP-ABE system as well as the other two systems in the same computational environment. The concrete parameters are chosen as follows.

A. Parameters

We implement the ABE schemes using a 224-bit MNT elliptic curve from the Stanford Pairing-Based Crypto (PBC) library [21]. For $\kappa = 80$ bits security parameter, we choose $\ell_{\mathsf{H}_0} = \ell_{\mathsf{H}} = 160$, and encapsulate a random 128-bit symmetric key ℓ_{SE}. In our scheme, we first hash the element C_T of group $\mathbb{G}_T$ to a random "seed" and then apply a pseudorandom number generator (e.g., AES scheme) to extend it to a 512-bit key K. It sufficiently guarantees that $\log|\mathcal{K}| - \ell_{\mathsf{H}} \geq \ell_{\mathsf{SE}} + 2\log 2^{\kappa}$.

B. Implementation

To capture the worst influence on the ciphertext size and decryption time by the complexity of access structures, each ciphertext is generated under "AND" access policies (A_1 AND A_2 AND ... AND A_N), where each A_i is an attribute. Then, we generate a pair of corresponding transformation key and decryption key that contain the N attributes necessary for decryption. This approach captures the worst case that involves all the ciphertext components in the decryption operation. Experiments are conducted on an Intel Core i5 processor with 8GB RAM running 32-bit Windows 7 operation system. All the three implementations are just slight modifications of the libfenc ABE library [22] which includes the Waters CP-ABE scheme. Decryption times are estimated by picking the average over 100 iterations. We provide the results on the performance of the three outsourced ABE schemes in Table II. The numbers (10, 20, ...) in the first line of the table denotes the number of key (or policy) attributes.

C. Discussion

From Table II, we observe that for the three schemes, almost all the ciphertext processing is outsourced to a cloud server and that the outsourced ciphertext size and decryption time increase linearly with the number of policy attributes, while the local ciphertext size and decryption time are kept small and constant. Our new scheme is verifiable and has nearly the same efficiency as the underlying non-verifiable scheme [8]. Though our scheme and the scheme in [9] are both verifiable, the former is much more efficiency in both ciphertext size and decryption time than the latter. Specifically, our scheme has half the ciphertext sizes in both server and client and is $2 \sim 4$ times faster in decryption compared with the scheme in [9].

VI. CONCLUSION

In this paper, we proposed a simple and generic method to convert any ABE scheme with non-verifiable outsourced decryption to an ABE scheme with verifiable outsourced decryption in the standard model. To concretely assess the performance of the new method, we presented an instantiation of our generic method based on Green et al.'s outsourced CP-ABE scheme without verifiability. We implemented our instantiation, Green et al.'s scheme [8] and Lai et al.'s verifiable outsourced scheme [9] on PC. Experiment results showed that our method is nearly optimal in the sense that it introduces minimal overhead in exchange for verifiability.

REFERENCES

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3494, R. Cramer, Ed. Berlin, Germany: Springer-Verlag, 2005, pp. 457–473.

[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2006, pp. 89–98.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy*, May 2007, pp. 321–334.

[4] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography* (Lecture Notes in Computer Science), vol. 6571, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Berlin, Germany: Springer-Verlag, 2011, pp. 53–70.

[5] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2013, pp. 463–474.

[6] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2007, pp. 456–465.

[7] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. de Panafieu, and C. Ràfols, "Attribute-based encryption schemes with constant-size ciphertexts," *Theoretical Comput. Sci.*, vol. 422, pp. 15–38, Mar. 2012.

[8] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Secur. Symp.*, 2011, p. 34.

[9] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.

[10] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 6223, T. Rabin, Ed. Berlin, Germany: Springer-Verlag, 2010, pp. 465–482.

[11] K.-M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 6223, T. Rabin, Ed. Berlin, Germany: Springer-Verlag, 2010, pp. 483–501.

[12] B. Chevallier-Mames, J.-S. Coron, N. McCullagh, D. Naccache, and M. Scott, "Secure delegation of elliptic-curve pairing," in *Smart Card Research and Advanced Application* (Lecture Notes in Computer Science), vol. 6035, D. Gollmann, J.-L. Lanet, and J. Iguchi-Cartigny, Eds. Berlin, Germany: Springer-Verlag, 2010, pp. 24–35.

[13] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. STOC*, 2009, pp. 169–178.

[14] C. Gentry and S. Halevi, "Implementing Gentry's fully-homomorphic encryption scheme," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 6632, K. G. Paterson, Ed. Berlin, Germany: Springer-Verlag, 2011, pp. 129–148.

[15] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in *Proc. CCS*, 1993, pp. 62–73.

[16] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 4, pp. 2201–2210, Aug. 2014. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6642027

[17] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Dept. Comput. Sci., Israel Inst. Technol., Haifa, Israel, 1996.

[18] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[19] V. Shoup, "Sequences of games: A tool for taming complexity in security proofs," Dept. Comput. Sci., New York Univ., New York, NY, USA, Tech. Rep. 2004/332, 2004. [Online]. Available: http://eprint.iacr.org/

[20] R. Canetti, H. Krawczyk, and J. B. Nielsen, "Relaxing chosen-ciphertext security," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 2729, D. Boneh, Ed. Berlin, Germany: Springer-Verlag, 2003, pp. 565–582.

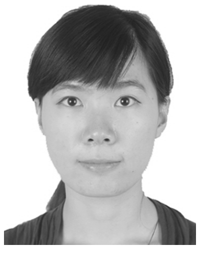
[21] B. Lynn. *The Stanford Pairing Based Crypto Library*. [Online]. Available: http://crypto.stanford.edu/pbc, accessed May 7, 2014.

[22] M. Green, A. Akinyele, and M. Rushanan. *libfenc: The Functional Encryption Library*. [Online]. Available: http://code.google.com/p/libfenc, accessed May 7, 2014.

Baodong Qin received the B.Sc. degree in information security and the M.Sc. degree in system analysis and integration from Shandong University, Jinan, China, in 2004 and 2007, respectively. He is currently pursuing the Ph.D. degree with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong university, China. He was a Research Assistant with the School of Information System, Singapore Management University, Singapore. His research interests include cryptography and provable security.

Robert H. Deng received the bachelor's degree from the National University of Defense Technology, China, and the M.Sc. and Ph.D. degrees from the Illinois Institute of Technology, USA. He was a Principal Scientist and the Manager with the Infocomm Security Department, Institute for Infocomm Research, Singapore. He has been with the Singapore Management University since 2004, where he is currently a Professor and an Associate Dean of the Faculty and Research with the School of Information Systems. He holds 26 patents and has over 200 technical publications in international conferences and journals in the areas of computer networks, network security, and information security. 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 has served as the General Chair, Program Committee Chair, and Program Committee Member of numerous international conferences. He is an Associate Editor of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING and *Security and Communication Networks Journal* (Wiley), and a member of Editorial Board of the *Journal of Computer Science and Technology* from the Chinese Academy of Sciences and the *International Journal of Information Security*.

Shengli Liu received the bachelor's, master's, and Ph.D. degrees from Xidian University, in 1995, 1998, and 2000, respectively, and the Ph.D. degree from the Technische Universiteit Eindhoven, The Netherlands. From 2000 to 2002, she continued her research on cryptography. She is currently a Professor with Shanghai Jiao Tong University. Her research interests include public key cryptosystems, and information-theoretic security.



Siqi Ma is currently pursuing the Ph.D. degree with the School of Information Systems, Singapore Management University, Singapore. Her research interests include cloud computing security and network security.