IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 16, 2019, accepted March 9, 2019, date of publication March 25, 2019, date of current version April 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907364

Generic Approach to Outsource the Decryption of Attribute-Based Encryption in Cloud Computing

BAODONG QIN[1,2] **AND DONG ZHENG**[1,3]

[1]National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China
[2]State Key Laboratory of Cryptology, Beijing 100878, China
[3]Westone Cryptologic Research Center, Beijing 100070, China

Corresponding author: Dong Zheng (zhengdong_xupt@sina.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872292, and in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2018JQ6007 and Grant 2018JZ6001.

ABSTRACT The notion of attribute-based encryption with outsourced decryption (OD-ABE) was proposed by Green, Hohenberger, and Waters. In OD-ABE, the ABE ciphertext is converted to a partially-decrypted ciphertext that has a shorter bit length and a faster decryption time than that of the ABE ciphertext. In particular, the transformation can be performed by a powerful third party with a public transformation key. In this paper, we propose a generic approach for constructing ABE with outsourced decryption from standard ABE, as long as the later satisfies some additional properties. Its security can be reduced to the underlying standard ABE in the selective security model by a black-box way. To avoid the drawback of selective security in practice, we further propose a modified decryption outsourcing mode so that our generic construction can be adapted to satisfying adaptive security. This partially solves the open problem of constructing an OD-ABE scheme, and its adaptive security can be reduced to the underlying ABE scheme in a black-box way. Then, we present some concrete constructions that not only encompass existing ABE outsourcing schemes of Green *et al.*, but also result in new selectively/adaptively-secure OD-ABE schemes with more efficient transformation key generation algorithm. Finally, we use the PBC library to test the efficiency of our schemes and compare the results with some previous ones, which shows that our schemes are more efficient in terms of decryption outsourcing and transformation key generation.

INDEX TERMS Attribute-based encryption, cloud computing, outsourcing, security model.

I. INTRODUCTION

Cloud storage is a new concept derived from the concept of cloud computing. It has become a very convenient way to store and share data among various users. Due to the emergence of many cloud security issues [1], almost all data is stored in the cloud in encrypted form. Traditionally, the data is encrypted by the owner to a specific target user, so it can only be opened by the target recipient. Many classical (public-key) encryption algorithms, such as RSA [2], have such functionality. However, in the cloud computing field, many applications want to share data with multiple users according to their roles. This usually requires the owner to encrypt data based on certain policies rather than a specified set of users. To achieve such encryption functionality, Sahai and Waters [3] initialized a new concept of attribute-based encryption (ABE). There are two types of ABE: Ciphertext-Policy (CP) [4] and Key-Policy (KP) [5]. In the CP-ABE (resp. KP-ABE) system, the user's key is associated with a set of attributes S (resp. an access structure $\mathbb{A}$) and each ciphertext is associated with an access structure $\mathbb{A}$ (resp. a set of attribute S), so that the key can decrypt the ciphertext only if $(S, \mathbb{A})$ satisfies some predicate function $f(\cdot, \cdot)$. So far, although there are many ABE schemes [4], [6]–[9], a common efficiency drawback of ABE is that ciphertext size and decryption time increase with the complexity of the access policy. For resource-constrained devices, such as mobile phones, the problem becomes even more worse. As showed in [10], for the CP-ABE scheme due to Waters [11], encrypting under a policy with 100 attributes (resp. 20 attributes) produces an ABE ciphertext with size of 25KB (resp. 5KB), while decrypting such ciphertext on an ARM processor requires approximately 30 seconds (resp. 6 seconds) of sustained computation.

To address the efficiency issues in existing ABE schemes, Green, Hohenberger and Waters [10] put forth a new framework for ABE system, called ABE with

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Chi Chen.

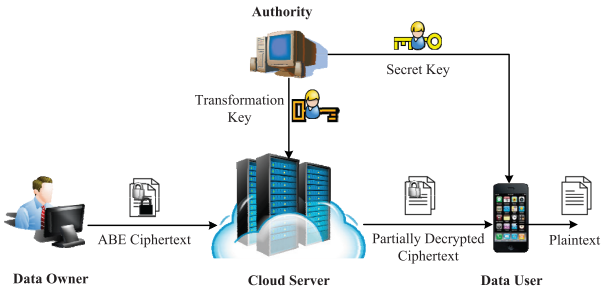

FIGURE 1. The framework of ABE with outsourced decryption.

outsourced decryption. Fig. 1 gives an overview of the new framework. There are four parties in this framework:

- *Authority:* It is the party who generates the system parameters, and data user's secret key and transformation key.
- *Data Owner:* It is the party who possesses data and encrypts them into cloud server.
- *Data User:* It is the party who has secret key to access encrypted data.
- *Cloud Server:* It is the party who stores data owner's encrypted data.

Compared to the original ABE decryption algorithm, the outsourcable ABE scheme splits the decryption key into two keys, so that the first key can be used to partially decrypt the ciphertext and the second key can be used to recover the message from the partially decrypted ciphertext. The first key is referred to as transformation key and can be publicly shared with a third computing party (e.g., a cloud storage and computing server). The attractive property of the transformation key is its application to the original ABE ciphertext, which produces a short El Gamal [12] type ciphertext (containing only two group elements). Therefore, the second key is just an El Gamal type secret key and is privately saved by the user. Since the transformation key can be publicly distributed, with the help of the cloud server, the new outsourced decryption framework significantly reduces the size of the original ABE ciphertext and the time of the final decryption operation.

Since the initial work of [10], many ABE schemes with outsourced decryption/encryption were proposed in recent years [13]–[18]. In addition, some works also support other special properties, including key escrowless [19], outsourced revocation [20]–[23] and verifiable outsourced decryption [24], [25]. Though there are many outsourcing ABE schemes, to the best of our knowledge, no work studies the general framework to construct OD-ABE schemes. Especially, it is very hard to construct a secure OD-ABE schemes in adaptive security model.

A. OUR CONTRIBUTION

In this paper, we initially study on the generic approach to outsource the decryption of ABE ciphertext, especially in the adaptive security model. The contribute of this paper is fourfold:

- First, we show that a standard ABE with some specific key-splitting properties can be used to realize decryption outsourcing. These properties are summarized as follows: the decryption key can be divided into two parts as the requirement of outsourcable ABE schemes. It requires the first part of the key to be obtained either in a private way (using the original ABE decryption key), or in a public way (using only the policy associated to the user). The private way will be used to generate the transformation key and the local user's secret key in the real world, while the public way is only used in the security proof. To guarantee efficiency, it also requires that the first part of the key significantly reduces the size of the ABE ciphertext and the time of decryption operation. We proved that the ABE scheme with the above properties results in an outsourceable ABE scheme in a black-box manner in the selective model (i.e., the adversary should commit to the challenger an encryption policy in advance).
- Second, in order to solve the practical issue in selective security model, we propose a modified decryption outsourcing mode, in which the user's transformation key and secret key are updated for each ciphertext. We show that our generic construction equipped with this modified decryption mode is provably secure in the adaptive security model, as long as the underlying ABE scheme is also adaptively secure.
  The technical idea employed in designing adaptively secure ABE with outsourcing is as follows: Though key updating, the user's transformation keys among different time periods are actually independent to each other. So, for any query on user's transformation key, the simulator can answer it without the corresponding decryption key. This is in contrast to the original decryption mode with outsourcing.
- Third, we propose two methods to split the decryption keys based on previous standard CP-ABE (without outsourcing) schemes of [11], [26]. Our results not only encompass the outsourcable ABE schemes proposed by Green et al., but also produce new OD-ABE schemes. In addition, the new schemes have more efficient transformation key generation algorithms than that of previous schemes.
- Fourth, we implemented our OD-ABE schemes using the PBC library [27] and compared the results with some previous schemes to quantitatively show the advantage of our generic approach in terms of transformation key generation.

B. ORGANIZATION

The rest of this paper is organized as follows. Section II recalls the cryptographic primitives and notions that will be used in this paper. Section III introduces the definition of OD-ABE and its security model. Section IV is our main work, including the key-splitting algorithm and the application to decryption outsourcing. The modified outsourced decryption

mode is presented in Section V. The instantiations of the key splitting algorithms are presented in Section VI. The implementation results are given in Section VII and the conclusion is given in Section VIII.

II. PRELIMINARIES

A. BILINEAR MAPS

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order p and let g be a generator of $\mathbb{G}$. A symmetric bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ has the following properties:

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

In addition, the group operation in $\mathbb{G}$ and the bilinear map e are both efficiently computable.

B. ACCESS STRUCTURE AND LINEAR SECRET-SHARING SCHEMES

Definition 1 (Access Structure [28]): Let $\{P_1, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{P_1,\ldots,P_n}$ is monotone if $\forall B, C$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$. An (monotone) access structure is a (monotone) collection $\mathbb{A}$ of non-empty subsets $\{P_1, \ldots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{P_1,\ldots,P_n} \setminus \{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets, otherwise they are called unauthorized sets.

We next recall the linear secret-sharing schemes (LSSS) of [28], that will be used to share secrets among a set of parties in our schemes.

Definition 2 (Linear Secret-Sharing Schemes (LSSS)): A scheme Π over a set of parities $\mathcal{P}$ (over $\mathbb{Z}_p$) is called a linear secret-sharing scheme if

1) For each party, the shares form a vector over $\mathbb{Z}_p$.
2) There exists a share-generating matrix M with ℓ rows and n columns, and an injective map ρ for Π that maps each row number i to the party $\rho(i)$. For any random vector $\vec{(v)} = (s, y_2, \ldots, y_n) \in \mathbb{Z}_p^n$, where s is the secret to be shared, we denote by $\lambda_i = M_i \cdot \vec{v}^{\mathsf{T}}$ the share belonging to party $\rho(i)$, where M_i is the i-th row vector of M.

It was shown in [28] that every linear secret sharing scheme with the above definition enjoys the linear reconstruction property, defined as follows: Suppose that Π is an LSSS for access structure $\mathbb{A}$ and $S \in \mathbb{A}$ be an authorized set and let I be the set of row numbers such that $\rho(i) \in S$. Then, there exists constants $\{w_i\}_{i \in I}$ such that, if $\{\lambda_i\}$ are valid shares of any secret s according to Π, then $\sum_{i \in I} w_i \cdot \lambda_i = s$. In addition, the constants $\{w_i\}$ can be found in time polynomial in the size of the sharing-generating matrix M. For any unauthorized set $S \notin \mathbb{A}$, the secret s is information-theoretically hidden from the parties in S.

III. ABE WITH OUTSOURCING

There are two types of ABE: ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE). Let $\mathbb{A}$ denotes an access structure and S denotes a set of attribute sets. For generality, we define (I_{key}, I_{enc}) as the inputs to the key generation algorithm and encryption algorithm respectively, and define a boolean function $f(I_{key}, I_{enc})$. For CP-ABE, we have $(I_{key}, I_{enc}) := (S, \mathbb{A})$ and $f(I_{key}, I_{enc}) := f(S, \mathbb{A})$, while for KP-ABE, we have $(I_{key}, I_{enc}) := (\mathbb{A}, S)$ and $f(I_{key}, I_{enc}) := f(\mathbb{A}, S)$. In either case, the function $f(I_{key}, I_{enc}) = 1$, if and only if $S \in \mathbb{A}$.

A CP-ABE (resp. KP-ABE) scheme with outsourced decryption consists of the following five (PPT) algorithms $(\mathsf{Setup}, \mathsf{Encrypt}, \mathsf{KeyGen}_{out}, \mathsf{Transform}, \mathsf{Decrypt}_{out})$.

- $(sp, msk) \leftarrow \mathsf{Setup}(\lambda, U)$. The setup algorithm takes as input a security parameter λ and an attribute universe description U. It outputs the system parameter sp and a master secret key msk. (The following algorithms may take the system parameter as an additional input. For the sake of simplicity, we omit it without explicitly explaination.)
- $CT \leftarrow \mathsf{Encrypt}(m, I_{enc})$. Then encryption algorithm takes as input a message m and an access structure (resp. attribute set) I_{enc}. It outputs a ciphertext CT.
- $(tk, sk) \leftarrow \mathsf{KeyGen}_{out}(msk, I_{key})$. The key generation algorithm takes as input the master secret key msk and an attribute set (resp. access structure) I_{key}. It outputs a ciphertext transformation key tk and a local secret (decryption) key sk.
- $CT'/\perp \leftarrow \mathsf{Transform}(tk, CT)$. The ciphertext transformation algorithm takes as input a transformation key tk for I_{key} and a ciphertext CT' encrypted under I_{enc}. It outputs the partially decrypted ciphertext CT' if $f(I_{key}, I_{enc}) = 1$, and the error symbol $\perp$ otherwise.
- $m/\perp \leftarrow \mathsf{Decrypt}_{out}(sk, CT')$. The decryption algorithm takes as input a secret key sk for I_{key} and a partially decrypted ciphertext CT' that was originally encrypted under I_{enc}. It outputs the message m if $f(I_{key}, I_{enc}) = 1$, and the error symbol $\perp$ otherwise.

For correctness, it requires that for all λ, U, I_{key}, I_{enc} and m, if $f(I_{key}, I_{enc}) = 1$, then

$$\mathsf{Decrypt}_{out}(sk, \mathsf{Transform}(tk, \mathsf{Encrypt}(m, I_{enc}))) = m$$

where $(sp, msk) \leftarrow \mathsf{Setup}(\lambda, U)$ and $(tk, sk) \leftarrow \mathsf{KeyGen}_{out}(msk, I_{key})$.

A. STANDARD ABE

The notion of *standard* attribute-based encryption can be similarly defined, if in the above definition, no such transformation key and ciphertext transformation algorithm exist.

B. SECURITY MODEL FOR ABE WITH OUTSOURCING

The security model is defined through an attack game played between the challenger and an adversary.

- *Setup:* The challenger runs the setup algorithm Setup and gives the system parameter, sp to the adversary.
- *Phase 1:* The adversary can access to the following two oracles for transformation keys and private keys repeatedly and adaptively. To answer the adversary's queries,

the challenger first initializes an empty table T and an integer j.

- $\mathsf{Create}(I_{key})$. The challenger sets $j := j+1$ and generates (tk, sk) via $\mathsf{KeyGen}_{out}(msk, I_{key})$. It gives tk to the adversary and stores the entry (j, I_{key}, tk, sk) to the table T.
- $\mathsf{Corrupt}(i)$. The challenger searches the table T with index i. If there exists such entry (i, I_{key}, tk, sk), it gives the secret key sk to the adversary. If no such entry exits, the challenger returns $\perp$. Suppose that the adversary queries the challenger repeatedly for private keys with attribute sets (resp. access structures) $I_{key}^{(1)}, \ldots, I_{key}^{(q_1)}$.

- *Challenge:* The adversary submits two equal length messages m_0 and m_1, and an access structure (resp. attribute set) I_{enc}^* so that $f(I_{key}^{(i)}, I_{enc}^*) \neq 1$ for all $i \in [1, q_1]$. The challenger flips a random bit $b \in \{0, 1\}$, and computes $CT^* = \mathsf{Encrypt}(m_b, I_{enc}^*)$. The ciphertext CT^* is given to the adversary.
- *Phase 2:* The adversary queries the challenger repeatedly for secret keys corresponding to attribute sets (resp. access structures) $I_{key}^{(q_1+1)}, \ldots, I_{key}^{(q)}$ with the restriction that $f(I_{key}^{(i)}, I_{enc}^*) \neq 1$ for all $i \in [q_1 + 1, q]$.
- *Guess:* The adversary outputs a guess b' for b.

The advantage of an adversary in this game is defined as $|\Pr[b' = b] - 1/2|$.

Definition 3 (Adaptive Security): An ABE scheme with outsourcing is adaptively secure against chosen-plaintext attacks (CPA-secure) if all PPT adversaries have a negligible advantage in the above attack game.

Definition 4 (Selective Security): We say an ABE scheme is selectively secure against chosen-plaintext attacks (selective CPA-secure), if the adversary commits to the challenger an access structure (resp. attribute set) I_{enc}^* at the beginning of the Setup phase.

IV. GENERIC CONSTRUCTION

Suppose that ABE=(ABE.Setup, ABE.Encrypt, ABE.KeyGen, ABE.Decrypt) be a standard CP-ABE (resp. KP-ABE) scheme. For any attribute set S (resp. access structure $\mathbb{A}$), suppose that the corresponding decryption key dk can be split into two parts tk and sk, such that $dk = tk \odot sk$ for some algebraic operation "$\odot$". We call tk and sk outsourced transformation key and local secret key respectively. Let $\mathcal{TK}$ and $\mathcal{SK}$ denote the transformation key space and the secret key space respectively. Then, we require that the standard ABE scheme satisfies the following five properties.

- *Property 1* (Private Evaluation): There exists an algorithm Priv that takes as input the decryption key dk, and outputs a transformation key $tk \in \mathcal{TK}$ and a secret key $sk \in \mathcal{SK}$, so that $dk = tk \odot sk$.
- *Property 2* (Public Evaluation): There exits an algorithm Pub that takes as input the access structure (or the attribute set) I_{key}, and outputs a $tk \in \mathcal{TK}$, which has the same distribution as the first output of Priv.
- *Property 3* (Indistinguishability): For all I_{key} and all $dk \leftarrow \mathsf{ABE.KeyGen}(I_{key})$, the first output (namely tk) of $\mathsf{Priv}(dk)$ and the output of $\mathsf{Pub}(I_{key})$ are statistically indistinguishable.
- *Property 4* (Correctness): For any ciphertext CT, if $f(I_{key}, I_{enc}) = 1$, there always exist two efficient algorithms Dec_{server} and Dec_{user}, such that

$$\mathsf{ABE.Decrypt}(dk, CT) = \mathsf{Dec}_{user}(sk, \mathsf{Dec}_{server}(tk, CT)).$$

For efficiency, we require that the local decryption time t_{user} proceeded by algorithm Dec_{user} is significantly less than the full decryption time t_{dec} proceeded by algorithm Decrypt, i.e., $t_{user} < t_{dec}$. In addition, we require that the size of the partially decrypted ciphertext by Dec_{server} is shorter than that of the original ABE ciphertext size.

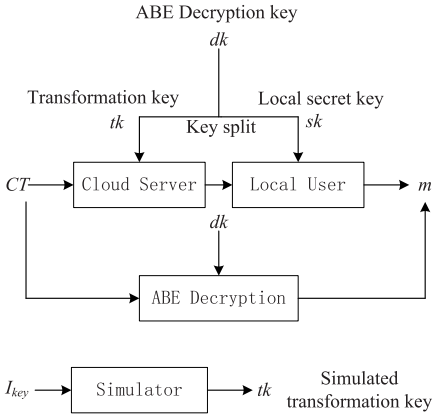

FIGURE 2. The idea of our generic construction.

Construction 1 (Generic construction of OD-ABE): Let ABE *be a standard ABE scheme supporting the above properties. Fig. 2 gives the main idea of the generic construction of ABE with outsourced decryption. The formal construction is described as follows.*

- $(sp, msk) \leftarrow \mathsf{Setup}(\lambda, U)$. *The setup algorithm is identical to that of the original ABE scheme. It runs* $\mathsf{ABE.Setup}(\lambda, U)$ *and outputs the system parameter sp and a master secret key msk.*
- $CT \leftarrow \mathsf{Encrypt}(m, I_{enc})$. *Then encryption algorithm is identical to that of the original ABE scheme. It runs* $\mathsf{ABE.Encrypt}(m, I_{enc})$ *and outputs the ciphertext CT.*
- $(tk, sk) \leftarrow \mathsf{KeyGen}_{out}(msk, I_{key})$. *It first runs the ABE key generation algorithm* $\mathsf{ABE.KeyGen}(msk, I_{key})$ *to obtain the corresponding decryption key dk. Then, it runs the private evaluation algorithm* $\mathsf{Priv}(dk)$ *to generate the outsourced transformation key tk and the local secret key sk. Finally, it outputs the the transformation key tk and the local secret key sk.*
- $CT'/\perp \leftarrow \mathsf{Transform}(tk, CT)$. *It runs the outsourced server decryption algorithm* $\mathsf{Dec}_{server}(tk, CT)$ *to generate a partially decrypted ciphertext* CT' *if* $f(I_{key}, I_{enc}) = 1$, *and the error symbol* $\perp$ *otherwise.*

- $m/\perp \leftarrow \mathsf{Decrypt}_{out}(sk, CT')$. *It runs the local user decryption algorithm* $\mathsf{Dec}_{user}(sk, CT')$ *with the secret key* sk *for* I_{key} *and a partially decrypted ciphertext* CT' *that was originally encrypted under* I_{enc} *to recover the message* m *if* $f(I_{key}, I_{enc}) = 1$, *and the error symbol* $\perp$ *otherwise.*

By the properties of the underlying ABE scheme, we have the following result.

Theorem 1: If the standard ABE scheme is (selectively) CPA-secure and satisfies ***Property 1-4****, then the ABE with outsourced decryption in Construction 1 is at least selectively CPA-secure.*

The correctness of the obtained OD-ABE scheme follows directly from the correctness of the underlying standard ABE scheme. We next prove its security and introduce a black box reduction to the security of the underlying standard ABE scheme. To answer the adversary's Create and/or Corrupt queries on I_{key}, our main trick is to obtain the full decryption key dk by querying the challenger of the standard ABE and then apply the private evaluation algorithm Priv to generate local user's secret key sk and the cloud server's transformation key tk. If I_{key} satisfies the challenge encryption access structure (or attribute set) I^*_{enc}, we apply the public evaluation algorithm Pub to generate the corresponding transformation key.

proof: Let $\mathcal{A}$ denote an adversary that wants to break the security of the ABE scheme with outsourced decryption and let $\mathcal{C}$ be the challenger of the underlying standard ABE scheme. We now construct an efficient algorithm Sim that breaks the selective CPA-security of the standard ABE scheme with the help of the adversary $\mathcal{A}$. The algorithm Sim works as follows.

- *Initialization:* The adversary $\mathcal{A}$ commits to Sim a challenge access structure (resp. attribute set) I^*_{enc} at the beginning of the setup algorithm. Sim passes it on to the challenger $\mathcal{C}$ of the standard ABE scheme.
- *Setup:* Sim passes I^*_{enc} on to the challenger $\mathcal{C}$ of the standard ABE scheme. The later will run the ABE setup algorithm ABE.Setup and return the system parameter sp to Sim. Sim passes it on to the adversary.
- *Phase 1:* The adversary can access to the following two oracles for transformation keys and private keys repeatedly and adaptively. To answer the adversary's queries, Sim first initializes an empty table T and an integer j.
  - Create(I_{key}). Sim sets $j := j+1$. If $f(I_{key}, I^*_{enc}) \neq 1$, Sim involves the ABE challenger $\mathcal{C}$ to generate the corresponding decryption key dk, and then applies the private evaluation algorithm Priv(dk) to generate the ciphertext transformation key tk and the local secret key sk. It stores the entry (j, I_{key}, tk, sk) to the table T. If $f(I_{key}, I^*_{enc}) = 1$, Sim involves the public evaluation algorithm Pub(I_{key}) to generate the corresponding transformation key tk and stores then entry $(j, I_{key}, tk, \star)$ to the table T. Finally, Sim gives tk to the adversary.
  - Corrupt(i). Sim searches the table T with index i. If there exists such entry (i, I_{key}, tk, sk), it gives the secret key sk to the adversary. Clearly, for any corruption query i, the I_{key} can not satisfy the challenge I^*_{enc}. Otherwise, the adversary can break the security of the OD-ABE scheme trivially.
- *Challenge:* Once the adversary submits two equal length messages m_0 and m_1. Sim passes them on to $\mathcal{C}$. The challenger $\mathcal{C}$ returns a challenge ciphertext $CT^* = \mathsf{ABE.Encrypt}(m_b, I^*_{enc})$ for some random bit $b \in \{0, 1\}$ to Sim. Sim passes the ciphertext CT^* on to the adversary $\mathcal{A}$.
- *Phase 2:* The adversary queries Sim repeatedly for secret keys corresponding to attribute sets (resp. access structures) $I^{(q_1+1)}_{key}, \ldots, I^{(q)}_{key}$ with the restriction that $f(I^{(i)}_{key}, I^*_{enc}) \neq 1$ for all $i \in [q_1 + 1, q]$. Sim answers them as in Phase 1.
- *Guess:* Eventually, $\mathcal{A}$ outputs a guess b' for b. Sim passes it on to $\mathcal{A}$.

Since the first output (namely tk) of the private evaluation algorithm Priv and the output of the public evaluation algorithm Pub are statistically indistinguishable, the above game is identical to that of the real CPA security game. So, the algorithm Sim successfully breaks the CPA security of the standard ABE scheme, if and only if $b' = b$. That is,

$$|\Pr[b' = b] - \frac{1}{2}| \leq |\Pr[\mathsf{Sim}\ success] - \frac{1}{2}| \leq \epsilon_{ABE}$$

where ϵ_{ABE} denotes the best advantage for any PPT algorithm to break the CPA security of the underlying standard ABE scheme. This completes the proof of Theorem 1.

V. HOW TO ACHIEVE ADAPTIVE SECURITY?

Recall that the generic security reduction of Theorem 1 only holds in selective security model. However, the selective security model is some what artificial, it is still meaningful to design a scheme in adaptive security model. In the previous security proof, when the adversary queries the oracle Create on I_{key}, the simulator has the following two ways to generate the transformation key:

- If $f(I_{key}, I^*_{enc}) = 1$, it involves the public evaluation algorithm Pub($\cdot$) to obtain the transformation key directly.
- If $f(I_{key}, I^*_{enc}) \neq 1$, it first involves the standard ABE challenger to obtain a full decryption key d, and then applies the private evaluation algorithm Priv($\cdot$) to generate the transformation key.

For adaptive security, as the simulator does not know the challenge access structure (resp. attribute set) I^*_{enc} in advance, it would be an obstacle to provide Theorem 1. Indeed, Green, Hohenberger and Waters in [10, Section 5.1]pointed out that if the adversary makes a query to some transformation key, the reduction algorithm will face a dilemma to generate the transformation key from the above two ways. We observe that this dilemma in some sense is caused by the transformation key being fixed. That is, the transformation key is created only

once. When it is created, the simulator must know whether the corresponding secret key can be queried (i.e., corrupted) or not. For example, if I_{key} does not satisfy the challenge encryption policy I^*_{enc}, the simulator must create a transformation key tk and know its corresponding secret key sk, as the adversary later may corrupt the secret key associated to the pre-created transformation key.

In this section, we propose a new framework for OD-ABE to alleviate the above issue. The new framework is identical to the one introduced in [10] (also see Section III) with the exception of the outsourced decryption mode. In the modified decryption mode, the data user refreshes his transformation key and secret key (rather than a fixed key pair) for each new ciphertext. Suppose that (tk, sk) is the current key pair for the data user. There is an algorithm $\mathsf{Update}(\cdot,\cdot)$ that takes as input (tk, sk) and outputs a new refreshed key pair (tk', sk'). Let $CT_1, CT_2, \cdots, CT_n$ be a sequence of ciphertexts. Fig. 3 depicts this new decryption mode for outsourcing.

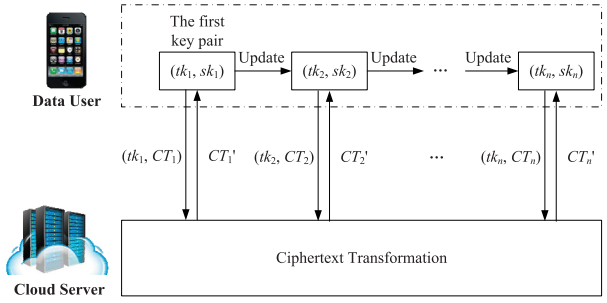

FIGURE 3. New outsouced decryption mode.

Due to key updating, it is reasonable to assume that during each decryption time period the adversary can only corrupt the current secret key. So, the two oracles (i.e., $\mathsf{Create}(I_{key})$ and $\mathsf{Corrupt}(i)$) in the previous security model for OD-ABE will be modified to the transformation key corruption oracle $\mathsf{CorruptTK}(I_{key})$ and the fully secret key (including both tk and sk) corruption oracle $\mathsf{CorruptSK}(I_{key})$, which are defined as follows:

- $\mathsf{CorruptTK}(I_{key})$: On input a query I_{key}, the oracle runs $\mathsf{KeyGen}_{out}(msk, I_{key})$ to generate a new key pair (tk, sk) and returns tk to the adversary.
- $\mathsf{CorruptSK}(I_{key})$: On input a query I_{key}, the oracle runs $\mathsf{KeyGen}_{out}(msk, I_{key})$ to generate a new key pair (tk, sk) and returns (tk, sk) to the adversary.

Now, we give the formal security model for the modified OD-ABE framework.

Security Model for ABE with Modified Outsourcing: The security model is also defined through an attack game played between the challenger and an adversary.

- *Setup:* The challenger runs the setup algorithm Setup and gives the system parameter, sp to the adversary.
- *Phase 1:* The adversary can access to the oracles $\mathsf{CorruptTK}(I_{key})$ and $\mathsf{CorruptSK}(I_{key})$ for transformation keys and secret keys repeatedly and adaptively. The challenger initializes an empty table T and adds the query I_{key} to T when the adversary queries the fully secret key oracle $\mathsf{CorruptSK}(I_{key})$.
- *Challenge:* The adversary submits two equal length messages m_0 and m_1, and an access structure (resp. attribute set) I^*_{enc} so that $f(I_{key}, I^*_{enc}) \neq 1$ for any $I_{key} \in T$. The challenger flips a random bit $b \in \{0, 1\}$, and computes $CT^* = \mathsf{Encrypt}(m_b, I^*_{enc})$. The ciphertext CT^* is given to the adversary.
- *Phase 2:* The adversary can repeatedly and adaptively query the oracles as in Phase 1, with the restriction that $f(I_{key}, I^*_{enc}) \neq 1$ when the adversary queries the fully secret key oracle with I_{key}.
- *Guess:* The adversary outputs a guess b' for b.

The advantage of an adversary in this game is defined as $|\Pr[b' = b] - 1/2|$.

Definition 5 (Adaptive Security): An ABE scheme with modified outsourcing is adaptively secure against chosen-plaintext attacks (CPA-secure) if all PPT adversaries have a negligible advantage in the above attack game.

To achieve adaptive security for the modified OD-ABE scheme, besides the properties proposed in Section IV, we additionally need the following property for standard ABE scheme:

- *Property 5* (Key Update): For any output (tk, sk) by algorithm Priv, there exists an efficient algorithm $\mathsf{Update}(\cdot,\cdot)$ that refreshes (tk, sk) to (tk', sk'), so that tk' has the same distribution as the output of the public evaluation algorithm Pub conditioned on (tk, sk).

For efficiency, the key update time t_{update} plus the local decryption time t_{user} should be significantly less than the full decryption time t_{dec}, i.e., $t_{update} + t_{user} < t_{dec}$.

By Property 1-5, we obtain the following result.

Theorem 2: If the underlying standard ABE scheme is adaptively CPA-secure and satisfies ***Property 1-5****, then the OD-ABE in Construction 1 equipped with the modified decryption mode in Fig. 3 is also adaptively CPA-secure.*

proof: The proof is almost identical to that of Theorem 1. The differences are summarized as follows:

- The adversary does not need to commit the challenge access structure (resp. attribute set) I^*_{enc} to the simulator Sim before the Setup phase.
- To answer the adversary's queries, the simulator will maintain a table T with elements of the form (I_{key}, tk, sk) and do it as follows.
  - When the adversary queries the transformation key on I_{key}, the simulator first searches the table T with index I_{key}: (1) If there exists entry (I_{key}, tk, sk), the simulator gives tk to the adversary and then involves the key update algorithm to update the key pair (tk, sk). Finally, it stores the refreshed entry (I_{key}, tk, sk) to table T. (2) If no such entry exists, the simulator always involves the public evaluation algorithm $\mathsf{Pub}(I_{key})$ to generate the corresponding transformation key tk and returns it to the adversary.

  - When the adversary queries the secret key on I_{key}, the simulator first searches the table T with index I_{key}: (1) If there exists such entry (I_{key}, tk, sk), it gives (tk, sk) to the adversary. It then involves the key update algorithm to update the key pair (tk, sk) and stores the refreshed entry (I_{key}, tk, sk) to table T. (2) If no such entry, the simulator always involves the original ABE challenger $\mathcal{C}$ to obtain the corresponding decryption key dk, and then applies the private evaluation algorithm $\mathsf{Priv}(dk)$ to generate the original key pair (tk, sk). The simulator returns (tk, sk) to the adversary and then involves the key update algorithm to update the key pair (tk, sk) and stores the refreshed entry (I_{key}, tk, sk) to table T.
- Once the adversary submits a challenge access structure (resp. attribute set) I^*_{enc} and two equal length messages m_0 and m_1, both previous and subsequent secret key queries on attribute set (resp. access structure) I_{key} should satisfy $f(I_{key}, I^*_{enc}) \neq 1$. The simulator passes them on to the underlying ABE challenger to generate the challenge ciphertext CT^*.

According to the property of the key update algorithm, the transformation keys generated by the key update algorithm and the public evaluation algorithm are statistically indistinguishable. So, breaking the adaptive CPA-security of the OD-ABE scheme equals to break the adaptive CPA-security of the underlying ABE scheme.

Remark: Though the modified decryption mode can be made to achieve adaptive security, it requires to frequently update user's transformation key and secret key. So, it introduces some additional computing operations to local decryption algorithm. It is still an open problem to prove adaptive security for normal outsourced decryption mode proposed by Green *et al.* [10].

VI. INSTANTIATIONS

In this section, we propose two methods for outsourcing the decryption of CP-ABE schemes. We believe these methods can be naturally extended to many KP-ABE schemes [5]–[7] for decryption outsourcing.

A. SELECTIVELY CPA-SECURE CONSTRUCTION

We improve the large universe construction of Waters CP-ABE scheme [11, Appendix C] to support decryption outsourcing, which is selectively CPA-secure under the decisional q-parallel BDHE assumption. We believe that these methods can be also applied to other CP-ABE schemes, such as [4], [7].

The CP-ABE of Waters [11]: The Waters CP-ABE for large universe is recalled as follows

- $\mathsf{Setup}(\lambda, U)$: The setup algorithm takes as input a security parameter λ and an attribute universe $U = \{0, 1\}^*$. It then chooses a finite group $\mathbb{G}$ of prime order p, a generator g and a bilinear map e from $\mathbb{G} \times \mathbb{G}$ to another finite group $\mathbb{G}_T$. In addition, it chooses a hash function $H : \{0, 1\}^* \to \mathbb{G}$. It also chooses two random exponents $\alpha, a \in \mathbb{Z}_p$. The authority sets the system parameter as $sp = (g, e(g, g)^\alpha, g^a, H)$ and sets the master secret key as $msk = g^\alpha$.
- $\mathsf{Encrypt}(m, I_{enc} = (M, \rho))$: Then encryption algorithm takes as input a message m and an LSSS access structure (M, ρ). The (injective) function ρ associates rows of M to attributes.
  Let M be an $\ell \times n$ matrix. The algorithm first chooses a random vector $\vec{v} = (s, y_2, \ldots, y_n) \in \mathbb{Z}_p^n$ that will be used to share the encryption exponent s. For $i = 1$ to ℓ, it computes $\lambda_i = M_i \cdot \vec{v}^{\mathsf{T}}$, where M_i is the i-th row of M. In addition, the algorithm picks ℓ random exponents $r_1, \ldots, r_\ell$, and computes the following ciphertext $CT = (C, C', \{C_i, D_i\}_{i\in[\ell]})$ where $C = m \cdot e(g, g)^{\alpha s}$, $C' = g^s$ and for $i \in [\ell]$, $C_i = g^{a\lambda_i} \cdot H(\rho(i))^{-r_i}$, $D_i = g^{r_i}$. Finally, it outputs the ciphertext CT along with a description of (M, ρ).
- $dk \leftarrow \mathsf{KeyGen}(msk, S)$: The key generation algorithm takes as input the master secret key msk and an attribute set S. It first chooses a random exponent $t \in \mathbb{Z}_p$ and then computes the decryption key $dk = (K, L, (K_x)_{x\in S})$, where $K = g^\alpha g^{at}$, $L = g^t$, and $\forall x \in S$, $K_x = H(x)^t$.
- $m/\perp \leftarrow \mathsf{Decrypt}(dk, CT)$: The decryption algorithm takes as input the decryption key dk for S and a ciphertext CT for access structure (M, ρ). Suppose that S satisfies the access structure (otherwise the algorithm outputs $\perp$ directly). Let $I \subset \{1, 2, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, let $\{w_i \in \mathbb{Z}_p\}_{i\in I}$ be a set of constants such that if λ_i are valid shares of any secret s according to M, then $\sum_{i\in I} w_i\lambda_i = s$. The decryption algorithm then computes

$$\frac{e(C', K)}{e(\Pi_{i\in I} C_i^{w_i}, L) \cdot \Pi_{i\in I} e(D_i^{w_i}, K_{\rho(i)})} = \frac{e(g, g)^{\alpha s} e(g, g)^{ast}}{\Pi_{i\in I} e(g, g)^{ta\lambda_i w_i}} = e(g, g)^{\alpha s}.$$

  Hence, the decryption algorithm can recover the message $m = C/e(g, g)^{\alpha s}$.

We present two methods to split the decryption key so that they satisfy the aforementioned Property 1-4. The two key splitting methods are described in Construction 2 and Construction 3 respectively.

Construction 2: The first key splitting method for selective security based on the CP-ABE scheme of Waters [11, Appendix C] is described as follows.

- $(tk, sk) \leftarrow \mathsf{Priv}(dk)$: *For any decryption key* $dk = (K = g^\alpha g^{at}, L = g^t, \{K_x = H(x)^t\}_{x\in S})$ *for some attribute set S, the private evaluation algorithm chooses a random exponent* $z \in \mathbb{Z}_p^*$, *and sets* $sk = z$ *and* $tk = (K', L', \{K'_x\}_{x\in S})$, *where*

$$K' = K^{1/z} = g^{\alpha/z} g^{a(t/z)}$$
$$L' = L^{1/z} = g^{t/z}$$
$$\{K'_x\}_{x\in S} = \{K_x^{1/z} = H(x)^{t/z}\}_{x\in S}.$$

- $tk \leftarrow \mathsf{Pub}(S)$: *For any attribute set S, the public evaluation algorithm chooses two random exponents $\beta, t \in \mathbb{Z}_p$, and sets $tk = (K' = g^{\beta} g^{at}, L' = g^t, \{K'_x = H(x)^t\}_{x \in S})$. This implicitly sets $sk = z = \alpha/\beta$. Clearly, this tk has the same distribution as the first output of* Priv *by the randomness of β and t.*
- $CT' \leftarrow \mathsf{Dec}_{server}(tk, CT)$: *For any ciphertext $CT = (C, C', \{C_i, D_i\}_{i \in [\ell]})$ for access structure (M, ρ), suppose that S satisfies the access structure (otherwise the algorithm outputs $\perp$ directly). As in the original decryption algorithm, let $I \subset \{1, 2, \ldots, \ell\}$ be the set $I = \{i : \rho(i) \in S\}$, and let $\{w_i \in \mathbb{Z}_p\}_{i \in I}$ be the set of constants such that if λ_i are valid shares of any secret s according to M, then $\sum_{i \in I} w_i \lambda_i = s$. The server decryption algorithm computes*

$$\frac{e(C', K')}{e(\Pi_{i \in I} C_i^{w_i}, L') \cdot \Pi_{i \in I} e(D_i^{w_i}, K'_{\rho(i)})} = \frac{e(g,g)^{(\alpha/z)s} e(g,g)^{as(t/z)}}{\Pi_{i \in I} e(g,g)^{(t/z) a \lambda_i w_i}} = e(g,g)^{(\alpha/z)s}.$$

Finally, the server decryption algorithm outputs $CT' = (T_0, T_1) = (C, e(g,g)^{(\alpha/z)s})$.
- $m \leftarrow \mathsf{Dec}(sk, CT')$: *For a secret key $sk = z$ and a partially decrypted ciphertext $CT' = (T_0, T_1)$, the user decryption algorithm computes $m = T_0 / T_1^z$. Since $T_0 = m \cdot e(g,g)^{\alpha s}$ and $T_1 = e(g,g)^{(\alpha/z)s}$, the correctness follows by $T_0 / T_1^z = m \cdot e(g,g)^{\alpha s} / (e(g,g)^{(\alpha/z)s})^z = m$.*

Construction 3: The second key splitting method for selective security based on the CP-ABE scheme of Waters [11, Appendix C] is described as follows.

- $(tk, sk) \leftarrow \mathsf{Priv}(dk)$: *For any decryption key $dk = (K = g^{\alpha} g^{at}, L = g^t, \{K_x = H(x)^t\}_{x \in S})$ for some attribute set S, the private evaluation algorithm chooses a random exponent $z \in \mathbb{Z}_p^*$, and sets $sk = z$ and $tk = (K', L', \{K'_x\}_{x \in S})$, where*

$$K' = K \cdot g^{-z} = g^{\alpha - z} g^{at}$$
$$L' = L = g^t$$
$$\{K'_x\}_{x \in S} = \{K_x = H(x)^t\}_{x \in S}.$$

- $tk \leftarrow \mathsf{Pub}(S)$: *For any attribute set S, the public evaluation algorithm chooses two random exponents $\beta, t \in \mathbb{Z}_p$, and sets $tk = (K' = g^{\beta} g^{at}, L' = g^t, \{K'_x = H(x)^t\}_{x \in S})$. This implicitly sets $sk = z = \alpha - \beta$. Clearly, this tk has the same distribution as the first output of* Priv *by the randomness of β and t.*
- $CT' \leftarrow \mathsf{Dec}_{server}(tk, CT)$: *For any ciphertext $CT = (C, C', \{C_i, D_i\}_{i \in [\ell]})$ for access structure (M, ρ), suppose that S satisfies the access structure (otherwise the algorithm outputs $\perp$ directly). As in the original decryption algorithm, let $I \subset \{1, 2, \ldots, \ell\}$ be the set $I = \{i : \rho(i) \in S\}$, and let $\{\omega \in \mathbb{Z}_p\}_{i \in I}$ be the set of constants such that if λ_i are valid shares of any secret s according to M, then $\sum_{i \in I} w_i \lambda_i = s$. The server decryption algorithm computes*

$$\frac{e(C', K')}{e(\Pi_{i \in I} C_i^{w_i}, L') \cdot \Pi_{i \in I} e(D_i^{w_i}, K'_{\rho(i)})} = \frac{e(g,g)^{(\alpha - z)s} e(g,g)^{ast}}{\Pi_{i \in I} e(g,g)^{t a \lambda_i \omega_i}} = e(g,g)^{(\alpha - z)s}.$$

Finally, the server decryption algorithm outputs $CT' = (T_0, T_1) = (C / e(g,g)^{(\alpha - z)s}, e(g, C'))$.
- $m \leftarrow \mathsf{Dec}(sk, CT')$: *For a secret key $sk = z$ and a partially decrypted ciphertext $CT' = (T_0, T_1)$, the user decryption algorithm computes $m = T_0 / T_1^z$. Since $T_0 = C / e(g,g)^{(\alpha - z)s} = m \cdot e(g,g)^{zs}$ and $T_1 = e(g, C') = e(g,g)^s$, the correctness follows by $T_0 / T_1^z = m \cdot e(g,g)^{zs} / (e(g,g)^s)^z = m$.*

Applying the first decryption key splitting method to our generic construction, we immediately obtain the CP-ABE outsourcing scheme proposed by Green *et al.* in [10, Figure 5]. Applying the second method, we obtain a new CP-ABE outsourcing scheme, which has the same efficiency as that of the first method in terms of local decryption operation and transformed ciphertext size. From Table 1, the second method require just one exponentiation to derive the transformation key from the full decryption key, while the first method requires $(2 + |S|)$ exponentiations over group $\mathbb{G}$.

B. ADAPTIVELY CPA-SECURE OD-ABE SCHEMES

We improve Lewko and Waters CP-ABE scheme [26] to support decryption outsourcing, which is adaptively CPA-secure under some complexity assumptions over a bilinear composite group, including the decisional q-parallel BDHE assumption.

The CP-ABE of Lewko and Waters [26]: The Lewko-Waters CP-ABE is actually an extension of the previous Waters CP-ABE from prime-order bilinear group to composite-order bilinear group. We recalled it as follows:

- $\mathsf{Setup}(\lambda, U)$: The setup algorithm chooses a bilinear group $\mathbb{G}$ of composite-order $N = p_1 p_2 p_3$ (3 distinct primes). Let $\mathbb{G}_{p_i}$ denote the subgroup of order p_i in $\mathbb{G}$. It then chooses random exponents $\alpha, a, b \in \mathbb{Z}_N$, and a random group element $g \in \mathbb{G}_{p_1}$. For each $x \in U$, it chooses a random exponent $h_x \in \mathbb{Z}_N$ and sets $H_x = g^{h_x}$. The authority sets the system parameter as $sp = (N, g, e(g,g)^{\alpha}, g^a, g^b, \{H_x\}_{x \in U})$ and sets the master secret key as $msk = (g^{\alpha}, g_3)$ where g_3 is a generator of $\mathbb{G}_{p_3}$.
- $\mathsf{Encrypt}(m, I_{enc} = (M, \rho))$: Then encryption algorithm takes as input a message m and an LSSS access structure (M, ρ). The (injective) function ρ associates rows of M to attributes.
  Let M be an $\ell \times n$ matrix. The algorithm first chooses a random vector $\vec{v} = (s, y_2, \ldots, y_n) \in \mathbb{Z}_N^n$ that will be used to share the encryption exponent s. For $i = 1$ to ℓ, it computes $\lambda_i = M_i \cdot \vec{v}^{\mathsf{T}}$, where M_i is the i-th row of M. In addition, the algorithm picks ℓ random exponents $r_1, \ldots, r_{\ell} \in \mathbb{Z}_N$, and computes the following ciphertext

$CT = (C, C', C'', \{C_i, D_i\}_{i\in[\ell]})$ where $C = m\cdot e(g,g)^{\alpha s}$, $C' = g^s$, $C'' = (g^b)^s$ and for $i \in [\ell]$, $C_i = g^{a\lambda_i} \cdot H_{\rho(i)}^{-r_i}$, $D_i = g^{r_i}$. Finally, it outputs the ciphertext CT along with a description of (M, ρ).

- $dk \leftarrow \mathsf{KeyGen}(msk, S)$: The key generation algorithm takes as input the master secret key msk and an attribute set S. It chooses random exponents $t, u \in \mathbb{Z}_N$, and random elements $R, R_1, R_2, \{R_x\}_{x\in S} \in \mathbb{G}_{p_3}$. The decryption key is $dk = (K, L_1, L_2, (K_x)_{x\in S})$, where $K = g^{\alpha} g^{at} g^{bu} R$, $L_1 = g^t R_1$, $L_2 = g^u R_2$, and $\forall x \in S$, $K_x = H_x^t R_x$.
- $m/\perp \leftarrow \mathsf{Decrypt}(dk, CT)$: The decryption algorithm takes as input the decryption key dk for S and a ciphertext CT for access structure (M, ρ). Suppose that S satisfies the access structure (otherwise the algorithm outputs $\perp$ directly). Let $I \subset \{1, 2, \ldots, \ell\}$ be defined as $I = \{i : \rho(i) \in S\}$. Then, let $\{w_i \in \mathbb{Z}_N\}_{i\in I}$ be a set of constants such that if λ_i are valid shares of any secret s according to M, then $\sum_{i\in I} w_i\lambda_i = s$. The decryption algorithm then computes

$$\frac{e(C', K)e(C'', L_2)^{-1}}{e(\Pi_{i\in I} C_i^{w_i}, L_1) \cdot \Pi_{i\in I} e(D_i^{w_i}, K_{\rho(i)})} = \frac{e(g,g)^{\alpha s} e(g,g)^{ast}}{\Pi_{i\in I} e(g,g)^{ta\lambda_i w_i}} = e(g,g)^{\alpha s}.$$

  Hence, the decryption algorithm can recover the message $m = C/e(g,g)^{\alpha s}$.

Similar to the key splitting method for selective security, we propose two decryption key splitting methods for adaptive security based on Lewko-Waters CP-ABE scheme. They are described in Construction 4 and Construction 5 respectively.

Construction 4: The first key splitting method for adaptive security based on the CP-ABE scheme of Lewko-Waters [26] is described as follows.

- $(tk, sk) \leftarrow \mathsf{Priv}(dk)$*: Suppose that the decryption key is* $dk = (K, L_1, L_2, (K_x)_{x\in S})$*, where* $K = g^{\alpha} g^{at} g^{bu} R$, $L_1 = g^t R_1$, $L_2 = g^u R_2$, *and* $\forall x \in S$, $K_x = H_x^t R_x$. *The private evaluation algorithm chooses a random exponent* $z \in \mathbb{Z}_p^*$, *and then sets* $sk = z$ *and* $tk = (K', L_1', L_2', \{K_x'\}_{x\in S})$, *where*

$$K' = K^{1/z} \quad L_1' = L_1^{1/z}$$
$$L_2' = L_2^{1/z} \quad \{K_x'\}_{x\in S} = \{K_x^{1/z}\}_{x\in S}.$$

- $tk \leftarrow \mathsf{Pub}(S)$*: For any attribute set* S, *the public evaluation algorithm chooses random exponents* $\beta, t, u \in \mathbb{Z}_N$, *and random elements* $R, R_1, R_2, \{R_x\}_{x\in S} \in \mathbb{G}_{p_3}$. *It sets* $tk = (K', L_1', L_2', \{K_x'\}_{x\in S})$, *where* $K' = g^{\beta} g^{at} g^{bu} R$, $L_1' = g^t R_1$, $L_2' = g^u R_2$, *and* $\forall x \in S$, $K_x' = H_x^t R_x$. *This implicitly sets* $sk = z = \alpha/\beta$. *Clearly, this tk has the same distribution as the first output of* Priv *by the randomness of* β, t, R, R_1, R_2 *and* $\{R_x\}_{x\in S}$.
- $(tk', sk') \leftarrow \mathsf{Update}(tk, sk)$*: For any key pair* $tk = (K', L_1', L_2', \{K_x'\}_{x\in S})$ *and* $sk = z$, *the key update algorithm first chooses random exponent* $z', t', u' \in \mathbb{Z}_N$ *and random elements* $R'', R_1'', R_2'', \{R_x''\}_{x\in S} \in \mathbb{G}_{p_3}$, *and then sets* $sk' = z \cdot z' \pmod N$ *and* $tk' = (K'', L_1'', L_2'', \{K_x''\}_{x\in S})$, *where*

$$K'' = K'^{1/z'} g^{at'} g^{bu'} R''$$
$$L_1'' = L_1'^{1/z'} g^{t'} R_1''$$
$$L_2'' = L_2'^{1/z'} g^{u'} R_2''$$
$$\{K_x''\}_{x\in S} = \{K_x'^{1/z'} H_x^{t'} R_x''\}_{x\in S}.$$

  By the randomness of z', t', u' *and* R'', R_1'', R_2'', $\{R_x''\}_{x\in S}$, *the above* tk' *has the same distribution as the output of the public evaluation algorithm.*
- $CT' \leftarrow \mathsf{Dec}_{server}(tk, CT)$*: For any ciphertext* $CT = (C, C', C'', \{C_i, D_i\}_{i\in[\ell]})$ *for access structure* (M, ρ), *suppose that* S *satisfies the access structure (otherwise the algorithm outputs* $\perp$ *directly). As in the original decryption algorithm, let* $I \subset \{1, 2, \ldots, \ell\}$ *be the set* $I = \{i : \rho(i) \in S\}$, *and let* $\{w_i \in \mathbb{Z}_N\}_{i\in I}$ *be the set of constants such that if* λ_i *are valid shares of any secret* s *according to* M, *then* $\sum_{i\in I} w_i\lambda_i = s$. *The server decryption algorithm computes*

$$\frac{e(C', K')e(C'', L_2')^{-1}}{e(\Pi_{i\in I} C_i^{w_i}, L_1') \cdot \Pi_{i\in I} e(D_i^{w_i}, K'_{\rho(i)})} = \frac{e(g,g)^{(\alpha/z)s} e(g,g)^{as(t/z)}}{\Pi_{i\in I} e(g,g)^{(t/z)a\lambda_i w_i}} = e(g,g)^{(\alpha/z)s}.$$

  Finally, the server decryption algorithm outputs $CT' = (T_0, T_1) = (C, e(g,g)^{(\alpha/z)s})$.
- $m \leftarrow \mathsf{Dec}(sk, CT')$*: For a secret key* $sk = z$ *and a partially decrypted ciphertext* $CT' = (T_0, T_1)$, *the user decryption algorithm computes* $m = T_0/T_1^z$. *Since* $T_0 = m \cdot e(g,g)^{\alpha s}$ *and* $T_1 = e(g,g)^{(\alpha/z)s}$, *the correctness follows by* $T_0/T_1^z = m \cdot e(g,g)^{\alpha s}/(e(g,g)^{(\alpha/z)s})^z = m$.

Construction 5: The second key splitting method for adaptive security based on the CP-ABE scheme of Lewko-Waters [26] is described as follows.

- $(tk, sk) \leftarrow \mathsf{Priv}(dk)$*: Suppose that the decryption key is* $dk = (K, L_1, L_2, (K_x)_{x\in S})$*, where* $K = g^{\alpha} g^{at} g^{bu} R$, $L_1 = g^t R_1$, $L_2 = g^u R_2$, *and* $\forall x \in S$, $K_x = H_x^t R_x$. *The private evaluation algorithm chooses a random exponent* $z \in \mathbb{Z}_p^*$ *and random elements* $R', R_1', R_2', \{R_x'\}_{x\in S} \in \mathbb{G}_{p_3}$, *and then sets* $sk = z$ *and* $tk = (K', L_1', L_2', \{K_x'\}_{x\in S})$, *where*

$$K' = K \cdot g^{-z} \quad L_1' = L_1$$
$$L_2' = L_2 \quad \{K_x'\}_{x\in S} = \{K_x\}_{x\in S}.$$

- $tk \leftarrow \mathsf{Pub}(S)$*: For any attribute set* S, *the public evaluation algorithm chooses random exponents* $\beta, t, u \in \mathbb{Z}_N$, *and random elements* $R, R_1, R_2, \{R_x\}_{x\in S} \in \mathbb{G}_{p_3}$. *It sets* $tk = (K', L_1', L_2', \{K_x'\}_{x\in S})$, *where* $K' = g^{\beta} g^{at} g^{bu} R$, $L_1' = g^t R_1$, $L_2' = g^u R_2$, *and* $\forall x \in S$, $K_x' = H_x^t R_x$. *This implicitly sets* $sk = z = \alpha - \beta \pmod N$. *Clearly, this tk has the same distribution as the first output of* Priv *by the randomness of* β, t, R, R_1, R_2 *and* $\{R_x\}_{x\in S}$.
- $(tk', sk') \leftarrow \mathsf{Update}(tk, sk)$*: For any key pair* $tk = (K', L_1', L_2', \{K_x'\}_{x\in S})$ *and* $sk = z$, *the key update*

TABLE 1. Efficiency comparison of CP-ABE outsourcing methods. Assume that the size of an attribute set is s and an LSSS access structure is associated with an $\ell \times n$ mathrix. Let P, E_G and E_T be the maximum time to compute a pairing, an exponentiation in $\mathbb{G}$ and an exponentiation in $\mathbb{G}_T$ respectively.

Schemes	Full CT Size	Full Dec. Ops	Trans. CT Size	Local Dec. Ops	Priv. Eval. Ops	Update Ops	Security Model	Group Type
Waters [11]	$\lvert\mathbb{G}_T\rvert + (1+2\ell)\lvert\mathbb{G}\rvert$	$\leq (2+\ell)P + 2\ell E_G$	-	-	-	-	Selective	Prime
Construction 2	$\lvert\mathbb{G}_T\rvert + (1+2\ell)\lvert\mathbb{G}\rvert$	$\leq (2+\ell)P + 2\ell E_G$	$2\lvert\mathbb{G}_T\rvert$	E_T	$(2+s)E_G$	-	Selective	Prime
Construction 3	$\lvert\mathbb{G}_T\rvert + (1+2\ell)\lvert\mathbb{G}\rvert$	$\leq (2+\ell)P + 2\ell E_G$	$2\lvert\mathbb{G}_T\rvert$	E_T	E_G	-	Selective	Prime
LW [26]	$\lvert\mathbb{G}_T\rvert + (2+2\ell)\lvert\mathbb{G}\rvert$	$\leq (3+\ell)P + 2\ell E_G$	-	-	-	-	Adaptive	Composite
Construction 4	$\lvert\mathbb{G}_T\rvert + (2+2\ell)\lvert\mathbb{G}\rvert$	$\leq (3+\ell)P + 2\ell E_G$	$2\lvert\mathbb{G}_T\rvert$	E_T	$(3+s)E_G$	$(7+2s)E_G$	Adaptive	Composite
Construction 5	$\lvert\mathbb{G}_T\rvert + (2+2\ell)\lvert\mathbb{G}\rvert$	$\leq (3+\ell)P + 2\ell E_G$	$2\lvert\mathbb{G}_T\rvert$	E_T	E_G	$(5+s)E_G$	Adaptive	Composite

algorithm first chooses random exponent $z', t', u' \in \mathbb{Z}_N$ *and random elements* $R'', R_1'', R_2'', \{R_x''\}_{x\in S} \in \mathbb{G}_{p_3}$, *and then sets* $sk' = z + z' \pmod N$ *and* $tk' = (K'', L_1'', L_2'', \{K_x''\}_{x\in S})$, *where*

$$\begin{aligned} K'' &= K' g^{-z'} g^{at'} g^{bu'} R'' \\ L_1'' &= L_1' g^{t'} R_1'' \\ L_2'' &= L_2' g^{u'} R_2'' \\ \{K_x''\}_{x\in S} &= \{K_x' H_x^{t'} R_x''\}_{x\in S}. \end{aligned}$$

By the randomness of z', t', u' *and* $R'', R_1'', R_2'', \{R_x''\}_{x\in S}$, *the above* tk' *has the same distribution as the output of the public evaluation algorithm.*

- $CT' \leftarrow \mathsf{Dec}_{server}(tk, CT)$: *For any ciphertext* $CT = (C, C', C'', \{C_i, D_i\}_{i\in[\ell]})$ *for access structure* (M, ρ), *suppose that* S *satisfies the access structure (otherwise the algorithm outputs* $\perp$ *directly). As in the original decryption algorithm, let* $I \subset \{1, 2, \ldots, \ell\}$ *be the set* $I = \{i : \rho(i) \in S\}$, *and let* $\{w_i \in \mathbb{Z}_N\}_{i\in I}$ *be the set of constants such that if* λ_i *are valid shares of any secret* s *according to* M, *then* $\sum_{i\in I} w_i\lambda_i = s$. *The server decryption algorithm computes*

$$\frac{e(C', K')e(C'', L_2')^{-1}}{e(\Pi_{i\in I} C_i^{w_i}, L_1') \cdot \Pi_{i\in I} e(D_i^{w_i}, K'_{\rho(i)})} = \frac{e(g,g)^{(\alpha-z)s} e(g,g)^{ast}}{\Pi_{i\in I} e(g,g)^{ta\lambda_i w_i}} = e(g,g)^{(\alpha-z)s}.$$

Finally, the server decryption algorithm outputs $CT' = (T_0, T_1) = (C/e(g,g)^{(\alpha-z)s}, e(g, C'))$.

- $m \leftarrow \mathsf{Dec}(sk, CT')$: *For a secret key* $sk = z$ *and a partially decrypted ciphertext* $CT' = (T_0, T_1)$, *the user decryption algorithm computes* $m = T_0/T_1^z$. *Since* $T_0 = m \cdot e(g,g)^{zs}$ *and* $T_1 = e(g,g)^s$, *the correctness follows by* $T_0/T_1^z = m \cdot e(g,g)^{zs}/(e(g,g)^s)^z = m$.

Applying them to our generic construction, we immediately obtain two adaptively secure CP-ABE scheme with outsourced decryption. These two schemes are almost identical, with the exception of the key update algorithm. Specifically, from Table 1, the second method is slightly more efficient than that of the first method.

VII. PERFORMANCE COMPARISON

Theoretical Results: Table 1 summarizes the group operations and ciphertext length of the previous two key split algorithms for selective security and adaptive security respectively. Especially, the table lists the number of group operations of the private evaluation algorithm Priv. For selective security, both of the two outsourcing methods do not require key updating. So, the local user only requires the local decryption operation, which is very efficient and independent of the number of user's attributes. For adaptive security, they require key updating for each decryption. So, besides local decryption algorithm, the key update algorithm requires linear number of exponent operations. Nevertheless, compared with the full decryption algorithm, it does not require any expensive pairing operations.

Experimental Results: To evaluate the efficiency of our method in practice, we implement our OD-ABE schemes as well as Green et al.'s OD-ABE scheme [10] using the PBC library [27] on a Windows 10 platform with 2.2GHz Intel Core i5-5200U CPU and 8GB Memory. For selective security, the pairing is chosen from a Type-A elliptic curve $y^2 = x^3 + x$ defined over a 512-bit prime filed. For adaptive security, we choose a Type-A1 elliptic curve, which uses the same equation, but has a composite-order field. (including three 512-bit primes). We choose "AND" access policy and increase the policy attributes and user's attributes from 10 to 100.

Figure 4 illustrates the computation cost of selectively CPA-secure CP-ABE schemes with outsourced decryption. The first scheme comes from Green et al., which is identical to the one obtained using our first key splitting method. The second scheme is ours, obtained using our second key splitting method. Both of them are based Waters CP-ABE scheme. Fig. 4(a) and Fig. 4(b) show that both schemes significantly outsource the ABE ciphertext to the cloud server. For the transformation key generation algorithm, we assume that each user already has a valid normal CP-ABE decryption key and then derives the ciphertext transformation key from it. Fig. 4(c) shows that to derive the ciphertext transformation key, our method runs in constant time while Green et al.'s method runs in linear time with respect to the number of user's attributes. Fig. 4(d) and Fig. 4(e) illustrate that user's local decryption is very faster than full ABE decryption. It requires less than 2.5 milliseconds and is independent of the user's number of attributes.

Figure 5 illustrates the computation cost of adaptively CPA-secure CP-ABE schemes with outsourced decryption. The first scheme is obtained using our first key splitting method, while the second scheme is obtained using our second key splitting method, based on Lewko-Waters CP-ABE scheme. Fig. 5(a) and Fig. 5(b) illustrate that the original ABE

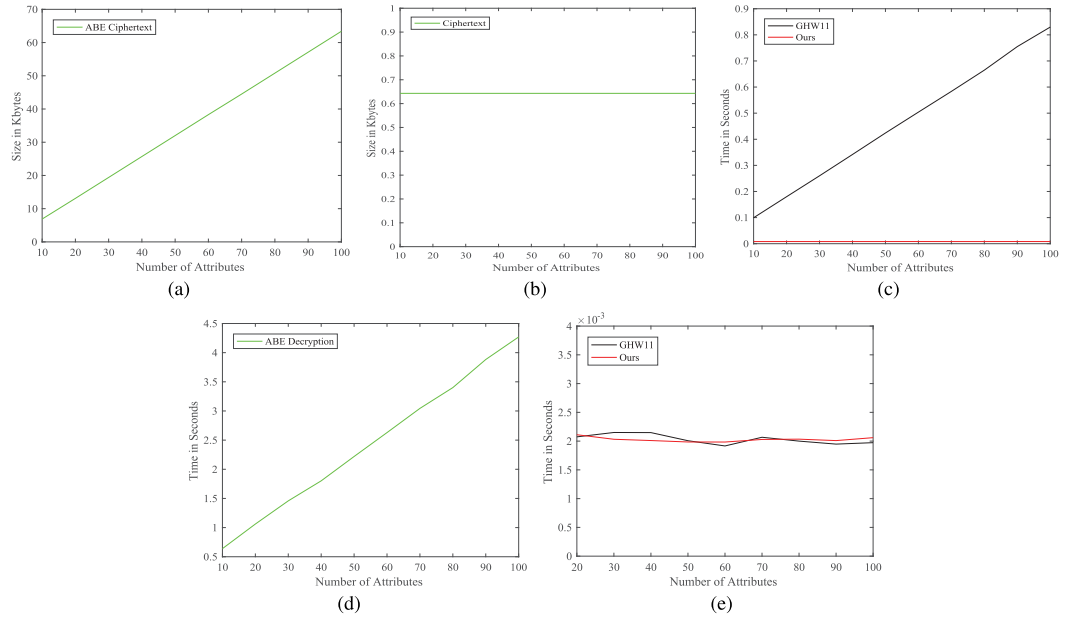

FIGURE 4. Experimental results for slectively CPA-secure CP-ABE with outsourced decryption. (a) ABE Ciphertext Size. (b) Transformed Ciphertext Size. (c) TK Generation Time. (d) Full ABE Decryption Time. (e) Final Decryption Time.

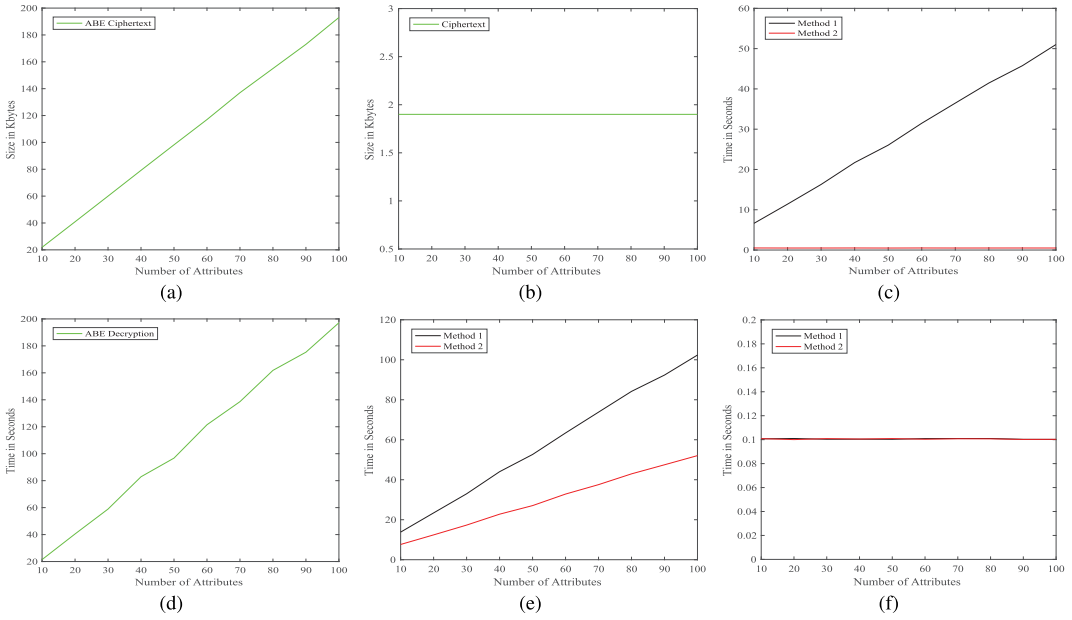

FIGURE 5. Experimental results for Adaptive CPA-secure CP-ABE with outsourced decryption. (a) ABE Ciphertext Size. (b) Transformed Ciphertext Size. (c) TK Generation Time. (d) Full ABE Decryption Time. (e) Key Update Time. (f) Final Decryption Time.

ciphertext length is linear to the number of policy attributes, but the transformed ciphertext length is constant and very short. For example, when the number of policy attributes is 100, the ABE ciphertext is almost 200 Kbytes, while the transformed ciphertext is only less than 2 Kbytes. As in our selectively CPA-secure schemes, Fig. 5(c) shows that the second TK generation algorithm is more efficient than the first one. For large attribute set, the second method runs in less than 1 seconds, while the first method requires more than 50 seconds. Fig. 5(d) depicts the normal ABE decryption time. To obtain adaptive security, the two methods both require transformation/secret key updating. Fig. 5(e) shows that they run in linear time with the number of user's attributes. Fortunately, they are faster than full ABE decryption. Specifically, the first method is nearly 2 times faster than full ABE decryption, while the second method is nearly 2 times faster than the first second. Fig. 5(f) shows that the final decryption needs only about 0.1 seconds.

VIII. CONCLUSION

This paper researched on the properties of a standard ABE scheme that can be used to achieve decryption outsourcing. The result requires the (publicly distributed) transformation key to be obtained either from the original decryption

key, or from just the policy associated to the decryption key. By this specific property, there exists a black-box construction of decryption outsourceble ABE from standard ABE in the selective security model. By updating the user's key pair, the adaptive security reduction also holds for our generic construction. In addition, the paper proposed two methods to split the decryption key satisfying the required properties. Experimental results showed the advantages of these two key splitting methods.

REFERENCES

[1] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," *J. Netw. Comput. Appl.*, vol. 79, pp. 88–115, Feb. 2017. doi: 10.1016/j.jnca.2016.11.027.

[2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978. doi: 10.1145/359340.359342.

[3] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology—EUROCRYPT*, vol. 3494, R. Cramer, Ed. Aarhus, Denmark: Springer, May 2005, pp. 457–473. doi: 10.1007/11426639_27.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Oakland, CA, USA, May 2007, pp. 321–334. doi: 10.1109/SP.2007.11.

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur. (CCS)*, A. Juels, R. N. Wright, and S. D. C. di Vimercati, Eds. ACM, Alexandria, VA, USA, Oct. 2006, pp. 89–98. doi: 10.1145/1180405.1180418.

[6] R. Ostrovsky, A. Sahai, and B. Waters, "Attribute-based encryption with non-monotonic access structures," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)* P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds., 2007, pp. 195–203. doi: 10.1145/1315245.1315270.

[7] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, A. Sadeghi, V. D. Gligor, and M. Yung, Eds., Nov. 2013, pp. 463–474. doi: 10.1145/2508859.2516672.

[8] M. Ambrona, G. Barthe, R. Gay, and H. Wee, "Attribute-based encryption in the generic group model: Automated proofs and new constructions," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds., 2017, pp. 647–664. doi: 10.1145/3133956.3134088.

[9] S. Agrawal and M. Chase, "FAME: fast attribute-based message encryption," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds., 2017, pp. 665–682. doi: 10.1145/3133956.3134014.

[10] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Conf. Secur.*, Aug. 2011, p. 34. [Online]. Available: http://static.usenix.org/events/sec11/tech/full_papers/Green.pdf

[11] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Public Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 6571, D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi, Eds. Berlin, Germany: Springer, 2011, pp. 53–70. doi: 10.1007/978-3-642-19379-8_4.

[12] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 196, G. R. Blakley and D. Chaum, Eds. Berlin, Germany: Springer, 1984, pp. 10–18. doi: 10.1007/3-540-39568-7_2.

[13] 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. 8, pp. 2201–2210, Aug. 2014. doi: 10.1109/TPDS.2013.271.

[14] S. Hohenberger and B. Waters, "Online/offline attribute-based encryption," in *Public-Key Cryptography—PKC* (Lecture Notes in Computer Science), vol. 8383, H. Krawczyk, Ed. Buenos Aires, Argentina, Springer, Mar. 2014, pp. 293–310. doi: 10.1007/978-3-642-54631-0_17.

[15] S. Lin, R. Zhang, H. Ma, and S. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2119–2130, Oct. 2015. doi: 10.1109/TIFS.2015.2449264.

[16] J. Li, F. Sha, Y. Zhang, X. Huang, and J. Shen, "Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length," *Secur. Commun. Netw.*, vol. 2017, Sep. 2017, Art. no. 3596205. doi: 10.1155/2017/3596205.

[17] R. Zhang, H. Ma, and Y. Lu, "Fine-grained access control system based on fully outsourced attribute-based encryption," *J. Syst. Softw.*, vol. 125, pp. 344–353, Mar. 2017. doi: 10.1016/j.jss.2016.12.018.

[18] C. Zuo, J. Shao, G. Wei, M. Xie, and M. Ji, "CCA-secure ABE with outsourced decryption for fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 730–738, Jan. 2018. doi: 10.1016/j.future.2016.10.028.

[19] Y. Chen, Q. Wen, W. Li, H. Zhang, and Z. Jin, "Generic construction of outsourced attribute-based encryption without key escrow," *IEEE Access*, vol. 6, pp. 58955–58966, 2018. doi: 10.1109/ACCESS.2018.2875070.

[20] H. Cui, R. H. Deng, Y. Li, and B. Qin, "Server-aided revocable attribute-based encryption," in *Computer Security—ESORICS* (Lecture Notes in Computer Science), vol. 9879, I. G. Askoxylakis, S. Ioannidis, S. K. Katsikas, and C. A. Meadows, Eds. Cham, Switzerland: Springer, 2016, pp. 570–587. doi: 10.1007/978-3-319-45741-3_29.

[21] B. Qin, Q. Zhao, D. Zheng, and H. Cui, "Server-aided revocable attribute-based encryption resilient to decryption key exposure," in *Cryptology and Network Security* (Lecture Notes in Computer Science), vol. 11261, S. Capkun and S. S. M. Chow, Eds. Cham, Switzerland: Springer, 2017, pp. 504–514. doi: 10.1007/978-3-030-02641-7_25.

[22] B. Qin, X. Liu, Z. Wei, and D. Zheng, "Space efficient revocable ibe for mobile devices in cloud computing," *SCIENCE CHINA Inf. Sci.*, to be published. [Online]. Available: http://engine.scichina.com/publisher/scp/journal/SCIS/doi/10.1007/s11432-018-9455-5?slug=abstract. doi: 10.1007/s11432-018-9455-5.

[23] B. Qin, Q. Zhao, and D. Zheng, "Bounded revocable and outsourceable ABE for secure data sharing," *Comput. J.*, vol. 61, no. 8, pp. 1259–1268, Aug. 2018. doi: 10.1093/comjnl/bxy063.

[24] B. Qin, R. H. Deng, S. Liu, and S. Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1384–1393, Jul. 2015. doi: 10.1109/TIFS.2015.2410137.

[25] 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. doi: 10.1109/TIFS.2013.2271848.

[26] A. B. Lewko and B. Waters, "New proof methods for attribute-based encryption: Achieving full security through selective techniques," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 7417, R. Safavi-Naini and R. Canetti, Eds. Berlin, Germany: Springer, 2012, pp. 180–198. doi: 10.1007/978-3-642-32009-5_12.

[27] B. Lynn and *et al.* (2013). *Pairing-based cryptography library*. [Online]. Available: https://crypto.stanford.edu/pbc/

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

BAODONG QIN received the B.S. and M.S. degrees from Shandong University, Jinan, China, in 2004 and 2007, respectively, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2015. From 2014 to 2015, he was a Research Assistant with the School of Information System, Singapore Management University (SMU), Singapore. He is currently an Associate Professor with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, China. His research interests include cryptography and information security.

DONG ZHENG received the Ph.D. degree from Xidian University, in 1999. He joined the School of Information Security Engineering, Shanghai Jiao Tong University. He is currently a Professor with the National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, China. His research interests include information theory, cryptography, and information security. He is a Senior Member of the Chinese Association for Cryptologic Research and a member of the Chinese Communication Society.