CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2015; **27**:2004–2027
Published online 13 October 2014 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3397

SPECIAL ISSUE PAPER

A ciphertext-policy attribute-based proxy re-encryption scheme for data sharing in public clouds

Kaitai Liang[1,*,†], Liming Fang[2], Duncan S. Wong[1] and Willy Susilo[3]

[1] *Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR*
[2] *Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*
[3] *School of Computer Science and Software Engineering, University of Wollongong, Northfields Avenue, Wollongong, NSW 2522, Australia*

SUMMARY

Ciphertext-policy attribute-based proxy re-encryption (CP-ABPRE) extends the traditional Proxy Re-Encryption (PRE) by allowing a semi-trusted proxy to transform a ciphertext under an access policy to another ciphertext with the same plaintext under a new access policy (i.e., *attribute-based re-encryption*). The proxy, however, learns nothing about the underlying plaintext. CP-ABPRE has many real world applications, such as fine-grained access control in cloud storage systems and medical records sharing among different hospitals. All the existing CP-ABPRE schemes are leaving chosen-ciphertext attack (CCA) security as an interesting open problem. This paper, for the first time, proposes a new CP-ABPRE scheme to tackle the problem. The new scheme supports attribute-based re-encryption with any monotonic access structures. Despite being constructed in the random oracle model, our scheme can be proven CCA secure under the decisional q-parallel bilinear Diffie–Hellman exponent assumption. Copyright © 2014 John Wiley & Sons, Ltd.

Received 14 July 2014; Revised 10 August 2014; Accepted 31 August 2014

KEY WORDS: proxy re-encryption; ciphertext-policy attribute-based proxy re-encryption; bilinear map; chosen-ciphertext security

1. INTRODUCTION

Introduced by Sahai and Waters [1], attribute-based encryption (ABE), which is a generalization of identity-based encryption (IBE), is able to effectively increase the flexibility of data sharing such that only users satisfying specific policy are allowed to access the data. ABE is applicable to many network applications, such as targeted broadcast and audit log applications [2]. In the literature, ABE mainly comes in two flavors: one is the key-policy ABE (KP-ABE), and the other is the ciphertext-policy ABE (CP-ABE). In the former, ciphertexts are labeled with attribute sets, and private keys are associated with access structures specifying which kinds of ciphertexts the receiver is able to decrypt. In the latter, however, the case is complementary. That is, ciphertexts are related to access structures, and attribute sets are assigned to private keys.

In what follows, we use medical data sharing as an example to illustrate the usage of CP-ABE and motivate our work as well. Consider the following scenario. A heart-disease patient Alice would like to find a clinic for regular medical examination via an on-line medical service agent (e.g.,

*Correspondence to: Kaitai Liang, Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR.
†E-mail: kliang4-c@my.cityu.edu.hk

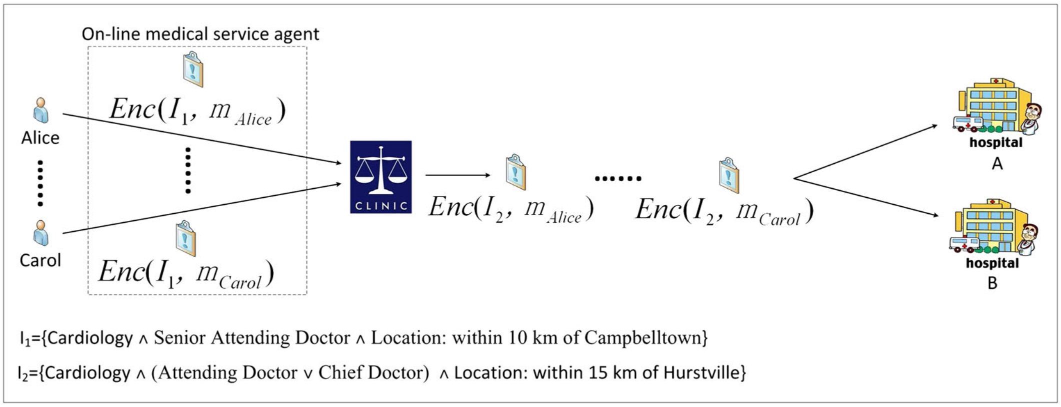
On-line medical service agent
Alice
$Enc(I_1, m_{Alice})$
Carol
$Enc(I_1, m_{Carol})$
CLINIC
$Enc(I_2, m_{Alice})$
......
$Enc(I_2, m_{Carol})$
hospital A
hospital B
I_1={Cardiology ∧ Senior Attending Doctor ∧ Location: within 10 km of Campbelltown}
I_2={Cardiology ∧ (Attending Doctor ∨ Chief Doctor) ∧ Location: within 15 km of Hurstville}

Figure 1. Traditional attribute-based encryption data sharing.

healthgrades[‡]). The clinic must be located within 10 km of Campbelltown, and the doctors (assigned for her treatment) of the clinic must be the senior attending doctors and be expert at cardiology. For simplicity, we denote Alice's requirements as $I_1 = \{Cardiology \wedge Senior\ Attending\ Doctor \wedge Location\ :\ within\ 10\ km\ of\ Campbelltown\}$. To protect the confidentiality of her record, Alice prefers to encrypt the record under I_1 (i.e., $Enc_{I_1}(m_{Alice})$) before sending to the on-line medical service agent ('the Agent'). The Agent (that knows I_1) then searches candidates satisfying I_1 in its database. Suppose there is a clinic matching I_1. The Agent then forwards Alice's ciphertext to the clinic. Note that the Agent cannot access Alice's data without knowledge of the private key (where the key is associated with an attribute set satisfying I_1).

Upon receiving Alice's ciphertext, the clinic satisfying I_1 is able to decrypt the ciphertext by using its private key so as to access the medical record. To keep trace of the medical record, the clinic may back up the ciphertext locally. When some cooperation is needed in the process of the treatment, Alice's medical record has to be sent to the cooperative hospitals. Suppose the hospitals should have the following requirements. They should be located within 15 km of Hurstville, and the doctors (assigned to the cooperation) of the hospitals should be the attending or chief doctors and must be expert at cardiology as well. We here denote the aforementioned requirements as $I_2 = \{Cardiology \wedge (Attending\ Doctor \vee Chief\ Doctor) \wedge Location\ :\ within\ 15\ km\ of\ Hurstville\}$, and suppose there are two hospitals, say hospital A and hospital B satisfying I_2.

In traditional data sharing, sharing Alice's medical record with A and B (without losing confidentiality), the clinic has to first recover m_{Alice} (from $Enc(I_1, m_{Alice})$ stored in the local server) and further encrypt the record under I_2 (i.e., $Enc(I_2, m_{Alice})$) before sending to hospitals A and B. However, if there are N patients who need to be cooperatively treated among the clinic, A and B, then the clinic will suffer from N pairs of encryption and decryption for their patients' records (Figure 1). This might be undesirable in practice because of high computational complexity.

To make data sharing be more efficient, proxy re-encryption (PRE) is proposed. Introduced by Mambo and Okamoto [3] and further defined by Blaze, Bleumer and Strauss [4], PRE extends the traditional public key encryption (PKE) to support the delegation of decryption rights. It allows a semi-trusted party called *proxy* to transform a ciphertext intended for Alice into another ciphertext of the same plaintext intended for Bob. The proxy, however, learns neither the decryption keys nor the underlying plaintext.

The PRE is a useful cryptographic primitive and has many applications, such as secure distributed files systems [5, 6], cloud data sharing [7], and email forwarding [4].

To date, PRE has been extended to adapt to different cryptographic settings. To achieve more flexibility on decryption rights delegation, many variants of PRE have been proposed, such as conditional PRE [8], identity-based PRE [9], and attribute-based PRE (ABPRE) [10]. This paper deals

[‡] http://www.healthgrades.com/.

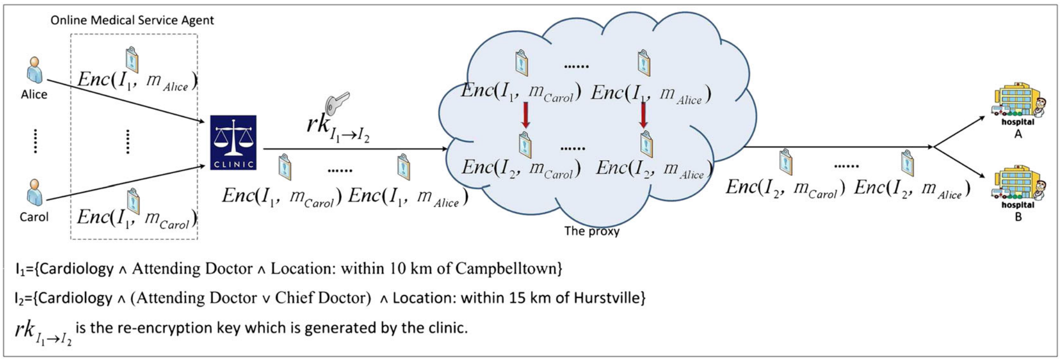
Online Medical Service Agent
Alice
Carol
$Enc(I_1, m_{Alice})$
$Enc(I_1, m_{Carol})$
CLINIC
$rk_{I_1 \to I_2}$
$Enc(I_1, m_{Carol})$ …… $Enc(I_1, m_{Alice})$
$Enc(I_1, m_{Carol})$ …… $Enc(I_1, m_{Alice})$
$Enc(I_2, m_{Carol})$ …… $Enc(I_2, m_{Alice})$
The proxy
$Enc(I_2, m_{Carol})$ …… $Enc(I_2, m_{Alice})$
hospital A
hospital B
I_1={Cardiology ∧ Attending Doctor ∧ Location: within 10 km of Campbelltown}
I_2={Cardiology ∧ (Attending Doctor ∨ Chief Doctor) ∧ Location: within 15 km of Hurstville}
$rk_{I_1 \to I_2}$ is the re-encryption key which is generated by the clinic.

Figure 2. Ciphertext-policy attribute-based proxy re-encryption.

with the case of ABPRE. Employing PRE in the context of ABE, Liang *et al.* [10] defined the notion of CP attribute-based proxy re-encryption (CP-ABPRE), and proposed the first concrete scheme, in which a proxy is allowed to transform a ciphertext under a specified access policy into the one under another access policy (i.e., *attribute-based re-encryption*).

Using the technology of CP-ABPRE, the medical data sharing discussed previously can be efficiently fulfilled as follows (Figure 2). The clinic, acting as a delegator, notifies a cloud (storage) server (acting as the proxy) that the hospitals satisfying I_2 (i.e., delegatees) should be granted the decryption rights of the data. The server then transforms the ciphertexts of the data under I_1 to the ones under I_2 by using a re-encryption key (e.g., $rk_{I_1 \to I_2}$, which is given by the clinic), such that A and B (satisfying I_2) can access the data. Note that the server does not learn the contents (of the data).

1.1. Applications of CP-ABPRE

We argue that CP-ABPRE explores the applications of PRE and has many real world applications, such as fine-grained data sharing in on-line medical service systems (e.g., Healthgrades[§]) and on-line payment systems (e.g., [11]). For example, in an on-line medical service system, a couple (living at Sydney) would like to find doctors with the following requirements to cure their child's bronchitis. Suppose the requirements are described as $I_3 = \{Paediatrician \wedge Bronchitis \wedge (Consultant \vee Registrar) \wedge Location : Downtown\ of\ Sydney\}$. The parent may encrypt the child's medical data under I_3 before uploading to the system. Note that because the system has no valid private key related to I_3, it cannot access the data. The system then forwards the ciphertext to the doctors satisfying I_3. Nevertheless, when one of the doctors goes out for medical trip, it is necessary to find some trustworthy substitutes to check the medical record. By employing CP-ABPRE, a doctor can first specify a new access policy, such as $I_4 = \{Paediatrician \wedge Bronchitis \wedge (Senior\ Registrar \vee Registrar)\}$, and then generates a re-encryption key (which can transform the ciphertext under I_3 into the one under I_4) for his or her proxy. When the doctor is absent, the proxy can convert the ciphertext of the data to the one that can be only decrypted by other doctors satisfying I_4.

The CP-ABPRE is also applicable to social network applications. For instance, a social network (e.g., LinkedIn[¶]) user, say Alice, might prefer to share her profile (e.g., educational details) with other system users under a specified access policy. Suppose the profile is encrypted under $P_1 = (`Region : United\ State'\ and\ `Occupation : student'\ and\ `Age : from\ 16\ to\ 25')$, and the encryption is already stored in the cloud such that the users satisfying $P_1$ can read the profile. Nonetheless, when Alice tries to 'link' herself with some companies for job applications, she might add a new access policy, like $P_2 = (`Region : all\ countries'\ and\ `People : HR\ or\ related\ officials'\ and\ `Field : software\ development')$, so as to additionally

[§] http://www.healthgrades.com/.
[¶] http://www.linkedin.com/.

enable the users matching P_2 to read the profile. Thus, a new encryption under P_2 is desirable. A naive solution for Alice to generate such an encryption is to download the ciphertext under P_1 from the cloud (if she does not store her profile locally), and next re-encrypt the profile under P_2 before uploading to the cloud. But Alice's workload here is increased, namely for each new policy, she has to manage the corresponding encryption(and decryption). Furthermore, if Alice is using some resource-limited devices, for example, mobile devices, which cannot afford the computational cost (incurred by the encryption and decryption), she cannot immediately share the profile with others unless some powerful devices (e.g., personal computer) are reachable.

However, Alice can efficiently share her profile with others by employing the technology of CP-ABPRE. Suppose the profile is encrypted under P_1 and stored in the cloud already. When Alice shares it with some human resource officials, she only needs to construct a re-encryption key from a set x of attributes to a new policy P_2, and next upload the key to the cloud, where x matches P_1. The cloud then re-encrypts the encryption under P_1 to another encryption under P_2 such that the officials satisfying P_2 can read the profile.

1.2. Open problems

Although CP-ABPRE is applicable to some network applications, it still leaves us interesting open problems in terms of security and functionality. All the existing CP-ABPRE schemes are only secure against *chosen-plaintext attacks* (CPA). The existence of CP-ABPRE with *chosen-ciphertext security* has been elusive. We argue that CPA security might be not sufficient enough in general protocol settings as it only achieves the very basic requirement from an encryption scheme, that is, secrecy against 'passive' eavesdroppers. When CP-ABPRE is implemented within a large protocol or system, which will be executed in an open network setting, a much wider array of attacks could be encountered with. For example, the adversary may control over the ciphertexts in the communication channel so as to either affect the decryption values or learn some partial information of the decryption results.

Chosen-ciphertext attack (CCA) security, however, allows the adversary to access the decryption oracle, that is, achieving the ability to read the underlying plaintexts related to the ciphertexts of its choice. This is able to preclude insider attacks. For example, a legitimate doctor of some hospital is able to acquire pairs of CP-ABPRE ciphertexts and plaintexts as previous knowledge. But the CCA security guarantees that he or she cannot gain any useful knowledge of the underlying plaintext of the challenge ciphertext after his or her retirement.

The CCA security also guarantees that if the adversary modifies given ciphertexts, then the modification for ciphertexts (affecting decryption results) can be detected. That is, even if the ciphertexts are modified and re-transferred to other receivers (whom are not the recipients specified by original sender), this misbehavior can be told by public verification. Therefore, it is desirable to propose a CCA-secure CP-ABPRE scheme in practice.

Another open problem left by the existing CP-ABPRE schemes is how to support more expressive access policy. As far as we know, all existing CP-ABPRE systems only support AND gates over attributes. However, an access policy might consist of AND, OR gates, and negative attributes in practice. Accordingly, it is desirable to design a CP-ABPRE scheme supporting expressive and flexible realization for access policy.

1.3. Our contributions

1.3.1. Difficulty of converting a CPA-secure CP-ABPRE to achieve CCA security. As stated in [10, 12], converting a CPA-secure CP-ABPRE (or PRE in general) scheme to obtain CCA security is a challenging open problem. One might think that some cryptographic primitives might be employed in the conversion, such as the CHK transformation [13]. The well-known CHK transformation can be used to convert a CPA-secure PKE scheme to be secure against CCA. The transformation, however, cannot be trivially employed in a CPA-secure PRE scheme to achieve CCA security. This is so because the CHK transformation is used to guarantee the integrity of ciphertexts, but at the same time, the transformation among ciphertexts in the PRE setting might affect the integrity. Trivially employing the CHK transformation in the context of PRE often yields a Replayable CCA [14] secure system, such as [15–17].

We here use an example to make a specific explanation. Suppose there is a CPA-secure CP-ABPRE scheme constructed in the standard model, and its original ciphertext is (A, B, C). In re-encryption, suppose the proxy generates (at least) a new component A', and outputs (A', B, C) as the re-encrypted ciphertext such that the valid delegatee can recover the plaintext from (A', B, C) by using his or her private key. Using the CHK transformation, the encryptor may make a signature D for (A, B, C) and output (K_v, A, B, C, D) as the original ciphertext, where K_v is the verification key corresponding to D. To validate D, the proxy has to keep A as an auxiliary output, that is, outputting (K_v, A, A', B, C, D). Despite the integrity of (A, B, C) can be verified by K_v and D, A' can be arbitrarily mutated by adversary. Note that the verification key K_v must be 'sealed' in ciphertext components such that an adversary cannot simply choose a new signing and verification key pair (K_v', K_s') and further make a new signature D' for (A, B, C). On the other hand, if only B and C are bound by D, then A's integrity cannot be guaranteed.

One might realize that the proxy might choose to sign A' in the CHK transformation as well such that the re-encrypted result is bound by signature. Nevertheless, this approach seems to be insensible. Suppose the proxy makes a signature D' for A' using a new signing key K_s', and outputs $(K_v, A, B, C, D, K_v', A', D')$ as the re-encrypted ciphertext, where K_v' is the verification key related to D'. An adversary may launch the following attacks: it first mutates A' as A'', next chooses (K_s'', K_v''), and then signs A'' in D''. The adversary finally outputs $(K_v, A, B, C, D, K_v'', A'', D'')$. Here, the verification for D'' will be passed, but the re-encryption result is mutated. The reason is that A' as a single component is loosely related to the original ciphertext and K_v'.

A naive solution for the problem is to request the proxy to not only encrypt A' under a new access policy, which associates with the delegatees' attributes, but also sign the new ciphertext in the CHK transformation. Nonetheless, this approach comes at a price that the communication bandwidth and the decryption complexity are both increased. Furthermore, it is possible for the proxy to suffer from either malicious attacks or invasion by an adversary in an open network such that the adversary might control the re-encryption and signature. Thus, this solution might be undesirable in terms of efficiency and security. In some privacy-preserving setting, that is, the proxy is not allowed to know the attributes of the corresponding delegatees; the solution is inappropriate as well.

Therefore, using the CHK transformation as a black box to turn the existing CPA-secure CP-ABPRE schemes to be secure against CCA is not trivial. In Section 4, we introduce an efficient solution to achieve CCA security in the context of CP-ABPRE.

In this work, we formalize the *selective access structure and chosen-ciphertext* (IND-sAS-CCA) security notion for CP-ABPRE systems. We state that it is the first time to define chosen-ciphertext security model for CP-ABPRE in the literature; and meanwhile, the notion discussed previously can be easily extended to the *adaptive access structure and chosen-ciphertext* (IND-aAS-CCA) security (note that we will show the details in Section 2.2). We consider the IND-sAS-CCA game into two different aspects: one is to allow the adversary to achieve an original ciphertext as the challenge ciphertext; the other is to allow the adversary to obtain a re-encrypted ciphertext as challenge. We refer to the security of the former and the latter as *IND-sAS-CCA security at original ciphertext* (i.e., IND-sAS-CCA-Or) and *IND-sAS-CCA security at re-encrypted ciphertext* (i.e., IND-sAS-CCA-Re), respectively. In this paper, we also show that the IND-sAS-CCA-Or security implies *selective collusion resistance*. Note that in [18], selective collusion resistance is also called as *selective master key security*.

The construction of a CP-ABPRE scheme with CCA security is an interesting open problem in the literature. This paper proposes the first single-hop unidirectional CP-ABPRE system to tackle the problem. It is also worth mentioning that all the existing CP-ABPRE schemes *only* support AND gates on (multi-valued) positive and negative attributes, while our scheme allows ciphertexts to be associated with any monotonic access formula. Although our scheme is constructed in the random oracle model, it can be proved that IND-sAS-CCA secure under the decisional q-parallel bilinear Diffie–Hellman exponent (q-parallel BDHE) assumption.

1.4. Related work

In 2005, Sahai and Waters [1] introduced the concept of ABE. There are two categories of ABE, KP-ABE, and CP-ABE. Goyal *et al.* [2] proposed the first KP-ABE, in which a ciphertext is related to a set of attributes, and each private key corresponds to an access policy over the attributes. The decryption can be fulfilled correctly if and only if the attribute set of the ciphertext satisfies the access policy on the decryptor's private key. Reversely, Bethencourt *et al.* [19] proposed CP-ABE where the ciphertext is associated with an access policy and the private key is related to an attribute set. Note that we here mainly focus on the review of CP-ABE. Later on, Cheung and Newport [20] proposed a provably secure CP-ABE scheme, which only supports AND gates over attributes. Goyal *et al.* [21] constructed a CP-ABE scheme but with large key size. Waters [22] designed efficient and expressive CP-ABE systems supporting any monotonic access structure. To convert one of the CP-ABE systems proposed in [22] to achieve fully security, Lewko *et al.* [23] adapted the dual system encryption technology to the ABE cryptographic setting. But their conversion yields some loss of expressiveness. Later, Lewko and Waters [24] introduced a new method to capture full security without jeopardizing the expressiveness by employing the selective proof technique into the dual system encryption technology. Attrapadung *et al.* [25] proposed an efficient CP-ABE for threshold access policy with constant-size ciphertexts. Later on, Waters [26] proposed the first deterministic finite automata-based functional encryption system in which access policy can be expressed by arbitrary-size regular language. Some variants of CP-ABE have been proposed in the literature, such as [27].

Following the introduction of decryption rights delegation by Mambo and Okamoto [3], Blaze *et al.* [4] formalized PRE and proposed a seminal bidirectional PRE scheme. After that, Ivan and Dodis [28] formalized the definitions of bidirectional and unidirectional proxy functions. In 2005, Ateniese *et al.* [5] proposed three unidirectional PRE schemes with CPA security. Later on, many classic PRE schemes (e.g., [15, 29–31]) have been proposed.

To employ PRE technology in the ABE cryptographic setting, Liang *et al.* [10] first defined CP-ABPRE, and further extended [20] to support PRE. This seminal CP-ABPRE system provides AND gates over positive and negative attributes. Later on, Luo *et al.* [18] proposed an extension of [10], in which their scheme supports policy with AND gates on multi-valued and negative attributes. To combine ABE with IBE by using PRE technique, Mizuno and Doi [32] proposed a special type of CP-ABPRE scheme. In this system, encryptions generated in the context of ABE can be converted to the ones being decrypted in the IBE cryptographic setting. The previously introduced systems, however, are CPA secure, and their access policies lack expressiveness because of supporting AND gates over attributes only. Thus, a CCA-secure CP-ABPRE scheme with more expressive access policy remains open. Recently, Liang *et al.* [33] proposed a system to solve the problem. This paper is a full version of [33], in which we explore some practical applications for the CP-ABPRE system with CCA security, revise the definition, security model and construction of the system, and present technical roadmap and full security proof.

We here compare our scheme with previous CP-ABPRE schemes, and summarize the comparison in terms of public/private key size, ciphertext/re-encryption key size, re-encryption cost, and properties, in Table I. We let f be the size of an access formula, A be the number of attributes on a user's private key, U be the number of all attributes defined in the system, mv be multi-valued attribute, $+$ be positive attribute and $-$ be negative attribute. Besides, we use c_e and c_p to denote the computational cost of an exponentiation and a bilinear pairing. To the best of our knowledge, our scheme is the first of its kind to achieve CCA security and to support any monotonic access formula (over attributes).

2. DEFINITIONS AND SECURITY MODELS

In this section, we concentrate on formulating the definition of CP-ABPRE systems. Before proceeding, we first review some notations used in our definition.

DOI: 10.1002/cpe

Table I. Comparison with [10, 18, 32].

Schemes	Public/private key size	Ciphertext/ ReKey size	Re-encryption cost	Selective model /CCA security	Attributes expression
[10]	$\mathcal{O}(U)/\mathcal{O}(U)$	$\mathcal{O}(U)/\mathcal{O}(U)$	$\mathcal{O}(U)c_p$	✓ /✗	AND:+ and −
[18]	$\mathcal{O}(U^2)/\mathcal{O}(U)$	$\mathcal{O}(U)/\mathcal{O}(U)$	$\mathcal{O}(U)c_p$	✓ /✗	AND:mv and −
[32]	$\mathcal{O}(U)/\mathcal{O}(U)$	$\mathcal{O}(U)/\mathcal{O}(U)$	$\mathcal{O}(1)c_e + \mathcal{O}(U)c_p$	✓ /✗	AND:+ and −
Ours	$\mathcal{O}(1)/\mathcal{O}(A)$	$\mathcal{O}(f)/\mathcal{O}(A)$	$\mathcal{O}(A)c_e + \mathcal{O}(A)c_p$	✓ /✓	Any monotonic

Definition 1
Access Structure [34]. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{AS} \subseteq 2^{\mathcal{P}}$ is monotone if $\forall B, C$: if $B \in \mathbb{AS}$ and $B \subseteq C$, then $C \in \mathbb{AS}$. An access structure (resp., monotonic access structure) is a collection (resp., monotone collection) $\mathbb{AS}$ of non-empty subsets of $\mathcal{P}$, that is, $\mathbb{AS} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$. The sets in $\mathbb{AS}$ are called the authorized sets, and the sets not in $\mathbb{AS}$ are called the unauthorized sets.

In the context of ABE, the role of the parties is taken by the attributes. The access structure $\mathbb{AS}$ contains all authorized sets of attributes. In this paper, we work on monotone access structures. As shown in [34], any monotone access structure can be represented by a linear secret sharing scheme.

Definition 2
Linear Secret Sharing Schemes (LSSS) [22]. A secret sharing scheme Π over a set of parties $\mathcal{P}$ is called linear (over $\mathbb{Z}_p$) if

- The shares for each party form a vector over $\mathbb{Z}_p$.
- There exists a matrix M with l rows and n columns called the share-generating matrix for Π. For all $i = 1, \ldots, l$, the ith row of M is labeled by a party $\rho(i)$, where ρ is a function from $\{1, \ldots, l\}$ to $\mathcal{P}$. When we consider the column vector $v = (s, r_2, \ldots, r_n)$, 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 $M \cdot v$ is the vector of l shares of the secret s according to Π. The share $(M \cdot v)_i$ belongs to party $\rho(i)$. For any unauthorized set, no such constants exist. We use LSSS matrix (M, ρ) to represent an access policy in this paper.

Note that every LSSS according to the previous definition achieves the *linear reconstruction* property [34]. Suppose Π is an LSSS for the access structure $\mathbb{AS}$. Let $S \in \mathbb{AS}$ (that is, S satisfies the access structure; we also denote this case as $S \models (M, \rho)$) be any authorized set, and let $I \subseteq \{1, 2, \ldots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$. There will exist constants $\{w_i \in \mathbb{Z}_p\}_{i \in I}$ such that $\sum_{i \in I} w_i \cdot \lambda_i = s$ if $\{\lambda_i\}$ are valid shares of any secret s according to Π. Note that as shown in [34], $\{w_i\}$ can be found (with knowledge of M and I) in time polynomial in the size of the share-generating matrix M.

2.1. Definition of CP-ABPRE

We review the definition of CP-ABPRE [10, 18] as follows.

Definition 3
A single-hop unidirectional CP-ABPRE scheme consists of the following seven algorithms:

(1) $(param, msk) \leftarrow Setup(1^k, \mathcal{U})$: on input a security parameter $k \in \mathbb{N}$ and an attribute universe $\mathcal{U}$, output the public parameters $param$ and a master secret key msk.
(2) $sk_S \leftarrow KeyGen(param, msk, S)$: on input $param$, msk and an attribute set S that describes the key, output a private key sk_S for S.
(3) $rk_{S \rightarrow (M', \rho')} \leftarrow ReKeyGen(param, sk_S, S, (M', \rho'))$: on input $param$, a private key sk_S and the corresponding attribute set S, and an access structure (M', ρ') for attributes over $\mathcal{U}$, output a re-encryption key $rk_{S \rightarrow (M', \rho')}$ that can be used to transform a ciphertext under

(M,ρ) to another ciphertext under (M',ρ'), where $S \models (M,\rho)$. Note (M,ρ) and (M',ρ') are disjoint. Suppose (M,ρ) and (M',ρ') are two access structures, for any attribute x satisfies (M,ρ), x does not satisfy (M',ρ'). For this case, from now on, we say that (M,ρ) and (M',ρ') are disjoint.

(4) $C_{(M,\rho)} \leftarrow Enc(param,(M,\rho),m)$: on input $param$, an access structure (M,ρ) for attributes over $\mathcal{U}$, and a message $m \in \{0,1\}^k$, output an original ciphertext $C_{(M,\rho)}$, which can be further re-encrypted. We assume the access structure is implicitly included in the ciphertext.

(5) $C^R_{(M',\rho')} \leftarrow ReEnc(param, rk_{S\rightarrow(M',\rho')}, C_{(M,\rho)})$: on input $param$, a re-encryption key $rk_{S\rightarrow(M',\rho')}$, and an original ciphertext $C_{(M,\rho)}$, output a re-encrypted ciphertext $C^R_{(M',\rho')}$ if $S \models (M,\rho)$ or a symbol $\perp$ indicating either $C_{(M,\rho)}$ is invalid or $S \not\models (M,\rho)$. Note that $C^R_{(M',\rho')}$ cannot be further re-encrypted.

(6) $m \leftarrow Dec(param, S, sk_S, C_{(M,\rho)})$: on input $param$, an attribute set S and its corresponding private key sk_S, and an original ciphertext $C_{(M,\rho)}$, output a message m if $S \models (M,\rho)$ or a symbol $\perp$ indicating either $C_{(M,\rho)}$ is invalid or $S \not\models (M,\rho)$.

(7) $m \leftarrow Dec_R(param, S', sk_{S'}, C^R_{(M',\rho')})$: on input $param$, an attribute set S' and its corresponding private key $sk_{S'}$, and a re-encrypted ciphertext $C^R_{(M',\rho')}$, output a message m if $S' \models (M',\rho')$ or a symbol $\perp$ indicating either $C^R_{(M',\rho')}$ is invalid or $S' \not\models (M',\rho')$.

For simplicity, we omit $param$ in the expression of the algorithm inputs in the rest of the paper.

Correctness: For any $k \in \mathbb{N}$, any attribute set S ($S \subseteq \mathcal{U}$) with its cardinality polynomial to k, any access structure (M,ρ) for attributes over $\mathcal{U}$ and any message $m \in \{0,1\}^k$, if $(param, msk) \leftarrow Setup(1^k, \mathcal{U})$, $sk_S \leftarrow KeyGen(msk, S)$, for all S used in the system, we have

$$Dec(S, sk_S, Enc((M,\rho),m)) = m;$$
$$Dec_R(S', sk_{S'}, ReEnc(ReKeyGen(sk_S, S, (M',\rho')), Enc((M,\rho),m))) = m,$$

where $S \models (M,\rho)$ and $S' \models (M',\rho')$.

2.2. Security models

In the following discussion, we define the security notions for CP-ABPRE systems. All existing notions for CP-ABPRE systems are only considered in the IND-sAS-CPA security model; later, we define a complete IND-sAS-CCA security game.

Definition 4

A single-hop unidirectional CP-ABPRE scheme is IND-sAS-CCA secure at original ciphertext if no Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$ can win the game later with non-negligible advantage. In the game, $\mathcal{C}$ is the game challenger, k and $\mathcal{U}$ are the security parameter and attribute universe.

(1) *Initialization.* $\mathcal{A}$ outputs a challenge access structure (M^*,ρ^*) to $\mathcal{C}$.

(2) *Setup.* $\mathcal{C}$ runs $Setup(1^k, \mathcal{U})$ and sends $param$ to $\mathcal{A}$.

(3) *Phase 1.* $\mathcal{A}$ is given access to the following oracles.

   (a) Private key extraction oracle $\mathcal{O}_{sk}(S)$: on input an attribute set S, $\mathcal{C}$ runs $sk_S \leftarrow KeyGen(msk, S)$ and returns sk_S to $\mathcal{A}$.

   (b) Re-encryption key extraction oracle $\mathcal{O}_{rk}(S, (M',\rho'))$: on input an attribute set S, and an access structure (M',ρ'), $\mathcal{C}$ returns $rk_{S\rightarrow(M',\rho')} \leftarrow ReKeyGen(sk_S, S, (M',\rho'))$ to $\mathcal{A}$, where $sk_S \leftarrow KeyGen(msk, S)$.

   (c) Re-encryption oracle $\mathcal{O}_{re}(S, (M',\rho'), C_{(M,\rho)})$: on input an attribute set S, an access structure (M',ρ'), and an original ciphertext $C_{(M,\rho)}$, $\mathcal{C}$ returns $C^R_{(M',\rho')} \leftarrow ReEnc(rk_{S\rightarrow(M',\rho')}, C_{(M,\rho)})$ to $\mathcal{A}$, where $rk_{S\rightarrow(M',\rho')} \leftarrow ReKeyGen(sk_S, S, (M',\rho'))$, $sk_S \leftarrow KeyGen(msk, S)$ and $S \models (M,\rho)$.

   (d) Original ciphertext decryption oracle $\mathcal{O}_{d2}(S, C_{(M,\rho)})$: on input an attribute set S and an original ciphertext $C_{(M,\rho)}$, $\mathcal{C}$ returns $m \leftarrow Dec(S, sk_S, C_{(M,\rho)})$ to $\mathcal{A}$, where $sk_S \leftarrow KeyGen(msk, S)$ and $S \models (M,\rho)$.

(e) Re-encrypted ciphertext decryption oracle $\mathcal{O}_{d1}(S', C^R_{(M',\rho')})$: on input an attribute set S' and a re-encrypted ciphertext $C^R_{(M',\rho')}$, $\mathcal{C}$ returns $m \leftarrow Dec_R(S', sk_{S'}, C^R_{(M',\rho')})$, where $sk_{S'} \leftarrow KeyGen(msk, S')$ and $S' \models (M', \rho')$.

Note if the ciphertexts issued to oracles $\mathcal{O}_{re}$, $\mathcal{O}_{d2}$ and $\mathcal{O}_{d1}$ are invalid, $\mathcal{C}$ simply outputs $\perp$. In this phase, the following queries are forbidden to issue:

- $\mathcal{O}_{sk}(S)$ for any $S \models (M^*, \rho^*)$; and
- $\mathcal{O}_{rk}(S, (M', \rho'))$ for any $S \models (M^*, \rho^*)$, and $\mathcal{O}_{sk}(S')$ for any $S' \models (M', \rho')$.

(4) *Challenge.* $\mathcal{A}$ outputs two equal length messages m_0 and m_1 to $\mathcal{C}$.
$\mathcal{C}$ returns $C^*_{(M^*,\rho^*)} = Enc((M^*, \rho^*), m_b)$ to $\mathcal{A}$, where $b \in_R \{0, 1\}$.

(5) *Phase 2.* $\mathcal{A}$ continues making queries as in Phase 1 except the following:

(a) $\mathcal{O}_{sk}(S)$ for any $S \models (M^*, \rho^*)$;
(b) $\mathcal{O}_{rk}(S, (M', \rho'))$ for any $S \models (M^*, \rho^*)$, and $\mathcal{O}_{sk}(S')$ for any $S' \models (M', \rho')$;
(c) $\mathcal{O}_{re}\left(S, (M', \rho'), C^*_{(M^*,\rho^*)}\right)$ for any $S \models (M^*, \rho^*)$, and $\mathcal{O}_{sk}(S')$ for any $S' \models (M', \rho')$;
(d) $\mathcal{O}_{d2}\left(S, C^*_{(M^*,\rho^*)}\right)$ for any $S \models (M^*, \rho^*)$; and
(e) $\mathcal{O}_{d1}\left(S', C^R_{(M',\rho')}\right)$ for any $C^R_{(M',\rho')}$, $S' \models (M', \rho')$, where $C^R_{(M',\rho')}$ is a derivative of $C^*_{(M^*,\rho^*)}$. As of [30], the derivative of $C^*_{(M^*,\rho^*)}$ is defined as follows.

(i) $C^*_{(M^*,\rho^*)}$ is a derivative of itself.
(ii) If $\mathcal{A}$ has issued a re-encryption key query on $(S, (M', \rho'))$ to obtain the re-encryption key $rk_{S\rightarrow(M',\rho')}$, and achieved $C^R_{(M',\rho')} \leftarrow ReEnc\left(rk_{S\rightarrow(M',\rho')}, C^*_{(M^*,\rho^*)}\right)$, then $C^R_{(M',\rho')}$ is a derivative of $C^*_{(M^*,\rho^*)}$, where $S \models (M^*, \rho^*)$.
(iii) If $\mathcal{A}$ has issued a re-encryption query on $\left(S, (M', \rho'), C^*_{(M^*,\rho^*)}\right)$ and obtained the re-encrypted ciphertext $C^R_{(M',\rho')}$, then $C^R_{(M',\rho')}$ is a derivative of $C^*_{(M^*,\rho^*)}$, where $S \models (M^*, \rho^*)$.

(6) *Guess.* $\mathcal{A}$ outputs a guess bit $b' \in \{0, 1\}$. If $b' = b$, $\mathcal{A}$ wins.

The advantage of $\mathcal{A}$ is defined as $\epsilon_1 = Adv^{IND-sAS-CCA-Or}_{CP-ABPRE,\mathcal{A}}(1^k, \mathcal{U}) = |Pr[b' = b] - \frac{1}{2}|$.

Remarks
The previous model can be extended to the IND-aAS-CCA-Or game by allowing $\mathcal{A}$ to output the challenge access structure (M^*, ρ^*) in the challenge phase. Meanwhile, there is no restriction for $\mathcal{A}$ in Phase 1. In addition, $\mathcal{C}$ will output the challenge ciphertext if the forbidden queries defined in Phase 1 of the previous game are never made.

The definition of IND-sAS-CCA-Re security is defined in an orthogonal manner as follows.

Definition 5
A single-hop unidirectional CP-ABPRE scheme is IND-sAS-CCA secure at re-encrypted ciphertext if the advantage $\epsilon_2 = Adv^{IND-sAS-CCA-Re}_{CP-ABPRE,\mathcal{A}}(1^k, \mathcal{U})$ is negligible for any PPT adversary $\mathcal{A}$ in the following experiment. Set $\mathcal{O} = \{\mathcal{O}_{sk}, \mathcal{O}_{rk}, \mathcal{O}_{re}, \mathcal{O}_{d2}, \mathcal{O}_{d1}\}$.

$$\begin{aligned}\epsilon_2 = \Big| Pr\Big[b' = b : ((M^*, \rho^*), State_1) \leftarrow \mathcal{A}(1^k); (param, msk). \\ \leftarrow Setup(1^k, \mathcal{U}); (m_0, m_1, (M, \rho), \\ State_2) \leftarrow \mathcal{A}^{\mathcal{O}}(param, State_1); b \in_R \{0, 1\}; C^{R*}_{(M^*,\rho^*)} \leftarrow ReEnc\big(rk_{S\rightarrow(M^*,\rho^*)}, C_{(M,\rho)}\big); \\ b' \leftarrow A^{\mathcal{O}}\left(C^{R*}_{(M^*,\rho^*)}, State_2\right)\Big] - \frac{1}{2}\Big|,\end{aligned}$$

where $State_1$ and $State_2$ are the state information, (M, ρ) and (M^*, ρ^*) are disjoint, (M^*, ρ^*) is the challenge access structure, $S \models (M, \rho)$, $rk_{S \to (M^*, \rho^*)} \leftarrow ReKeyGen(sk_S, S, (M^*, \rho^*))$, $C_{(M,\rho)} \leftarrow Enc((M, \rho), m_b)$, $\mathcal{O}_{sk}, \mathcal{O}_{rk}, \mathcal{O}_{re}, \mathcal{O}_{d2}, \mathcal{O}_{d1}$ are the oracles defined in Definition 4. However, these oracles are restricted by the following constraints. For $\mathcal{O}_{sk}$, the query on S is forbidden to issue for any $S \models (M^*, \rho^*)$. For $\mathcal{O}_{rk}$, it works as in the IND-sAS-CCA-Or game. $\mathcal{O}_{re}$ will output $\perp$ if $\mathcal{A}$ queries any invalid original ciphertext or any re-encrypted ciphertext. There is no restriction for $\mathcal{O}_{d2}$ except that the oracle will reject any invalid original ciphertext. If $\mathcal{A}$ queries to $\mathcal{O}_{d1}$ on $\left(S, C^{R*}_{(M^*, \rho^*)}\right)$ or invalid any re-encrypted ciphertext, the oracle outputs $\perp$, where $S \models (M^*, \rho^*)$.

Remarks
In Definition 5, $\mathcal{O}_{rk}$ must follow the constraints defined in Definition 4. This is necessary because in selective access structure model, the challenger cannot construct a valid private key for any $S \models (M^*, \rho^*)$. Thus, the re-encryption key $rk_{S \to (M', \rho')}$ has to be randomly generated, where $\mathcal{A}$ is allowed to query $\mathcal{O}_{sk}(S')$ for any $S' \models (M', \rho')$. If this re-encryption key is given to $\mathcal{A}$, then $\mathcal{A}$ can distinguish the simulation from the real attack. This is so because $\mathcal{A}$ can verify whether such a re-encryption key is valid or not as follows: $\mathcal{A}$ first generates a ciphertext of a chosen message m under (M^*, ρ^*), re-encrypts the ciphertext using the re-encryption key, and then decrypts the re-encrypted ciphertext by using $sk_{S'}$. If the decryption outputs m, then the re-encryption key is valid. Thus, this kind of re-encryption key cannot be sent to $\mathcal{A}$. As a result, to generate the corresponding re-encrypted ciphertext, $\mathcal{O}_{re}$ must be provided for $\mathcal{A}$. Note that Definition 5 can be regarded as a weaker notion when compared with the (adaptive) re-encrypted ciphertext security model (e.g., [12]).

The previous model can be also extended to the IND-aAS-CCA-Re security game by allowing $\mathcal{A}$ to output (M^*, ρ^*) in the challenge phase. Thus, there is no restriction for $\mathcal{A}$ to query $\mathcal{O}_{rk}$ such that $\mathcal{O}_{re}$ is unnecessary.

We now proceed to the *selective* collusion resistance for CP-ABPRE systems. Like collusion resistance defined in traditional PRE, this security notion also guarantees that a dishonest proxy cannot compromise the entire private key of the delegator when colluding with the corresponding delegatee. However, it requires an adversary to output an attribute set that it wishes to attack before the setup phase. The selective collusion resistance model can be defined in an identical way introduced in [10, 18]; we hence omit the details here. Instead, we prefer to show that the IND-sAS-CCA-Or security already implies the selective collusion resistance.

Theorem 1
Suppose a single-hop unidirectional CP-ABPRE scheme is IND-sAS-CCA-Or secure, then it is selective collusion resistant as well.

Proof
In the IND-sAS-CCA-Or security game, $\mathcal{A}$ can achieve the following re-encryption keys from $\mathcal{O}_{rk}$: $rk_{S \to (M', \rho')}$ and $rk_{S' \to (M'', \rho'')}$, where $S \models (M^*, \rho^*)$ and $S' \models (M', \rho')$. Following the restrictions defined in the game, $\mathcal{A}$ cannot query the private key $sk_{S'}$ for any $S' \models (M', \rho')$ (recall that $\mathcal{A}$ is forbidden to query any private key sk_S for any $S \models (M^*, \rho^*)$ as well) but the private key $sk_{S''}$ for any $S'' \models (M'', \rho'')$.

Suppose an IND-sAS-CCA-Or secure CP-ABPRE scheme is not selective collusion resistant. Then $\mathcal{A}$ is able to compromise the private key $sk_{S'}$ with knowledge of $rk_{S' \to (M'', \rho'')}$ and $sk_{S''}$. Using $rk_{S \to (M', \rho')}$, $\mathcal{A}$ can re-encrypt the challenge ciphertext $C^*_{(M^*, \rho^*)}$ to $C^R_{(M', \rho')}$. $\mathcal{A}$ then decrypts the re-encrypted ciphertext by using $sk_{S'}$ such that it can output the value of the bit b. This contradicts the IND-sAS-CCA-Or security.

This completes the proof of Theorem 1. □

3. PRELIMINARIES

We first give a brief review of bilinear maps and the decisional q-parallel BDHE assumption, and next introduce the target collision resistance hash function.

Bilinear Maps. Let $BSetup$ denote an algorithm that, on input the security parameter 1^k, outputs the parameters for a bilinear map as $(p, g, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups with prime order $p \in \Theta(2^k)$ and g is a generator of $\mathbb{G}$. The efficient mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ has three properties: (1) *bilinearity*: for all $g \in \mathbb{G}$ and $a, b \in_R \mathbb{Z}_p^*$, $e(g^a, g^b) = e(g, g)^{ab}$; (2) *non-degeneracy*: $e(g, g) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the unit of $\mathbb{G}_T$; and (3) *computability*: e can be efficiently computed.

Definition 6
Decisional q-parallel BDHE Assumption [22]. Given a tuple y=

$$g, g^s, g^a, \ldots, g^{a^q}, g^{a^{q+2}}, \ldots, g^{a^{2q}} \forall_{1 \le j \le q}\ g^{s \cdot b_j}, g^{a/b_j}, \ldots, g^{a^q/b_j}, g^{a^{q+2}/b_j}, \ldots, g^{a^{2q}/b_j}$$
$$\forall_{1 \le j,k \le q, k \neq j}\ g^{a \cdot s \cdot b_k/b_j}, \ldots, g^{a^q \cdot s \cdot b_k/b_j},$$

and $T \in \mathbb{G}_T$, the decisional q-parallel BDHE problem is to decide whether $T = e(g, g)^{a^{q+1} \cdot s}$, where $a, s, b_1, \ldots, b_q \in_R \mathbb{Z}_p$ and g is a generator of $\mathbb{G}$. Define $Adv_{\mathcal{A}}^{D-q-parallelBDHE} = |Pr[\mathcal{A}(y, e(g, g)^{a^{q+1} \cdot s}) = 0] - Pr[\mathcal{A}(y, T) = 0]|$ as the advantage of adversary $\mathcal{A}$ in winning the decisional q-parallel BDHE problem. We say that the decisional q-parallel BDHE assumption holds in $(\mathbb{G}, \mathbb{G}_T)$ if no PPT algorithm has non-negligible advantage.

Target Collision Resistant Hash Function. Target Collision Resistant (TCR) hash function was introduced by Cramer and Shoup [35]. A TCR hash function H guarantees that given a random element x, which is from the valid domain of H, a PPT adversary A cannot find $y \neq x$ such that $H(x) = H(y)$. We let $Adv_{H,A}^{TCR} = Pr[(x, y) \leftarrow \mathcal{A}(1^k) : H(x) = H(y), x \neq y, x, y \in DH]$ be the advantage of A in successfully finding collisions from a TCR hash function H, where DH is the valid input domain of H, k is the security parameter. If a hash function is chosen from a TCR hash function family, $Adv_{H,A}^{TCR}$ is negligible.

4. A NEW CP-ABPRE SCHEME WITH CCA SECURITY

In this section, we construct a new CP-ABPRE scheme in the random oracle model with CCA security. Prior to proposing the scheme, we first introduce some intuition behind our construction.

4.1. Intuition of our construction

We choose Waters ABE (the most efficient construction proposed in [22]) as a basic building block of our scheme with the following reasons. The construction of Waters ABE scheme enables us to convert the scheme to be an ABE Key Encapsulation in the random oracle model. Specifically, in our construction, a content key that is asymmetrically encrypted under an access policy is used to hide a message in a symmetric way. Furthermore, Waters ABE scheme leverages LSSS to support any monotonic access formula for data sharing. It is a desirable property for CP-ABPRE systems when being implemented in practice. In addition, Waters ABE scheme limits the size of ciphertext to be linear in the size of formula that helps us relieve the communication cost incurred by the delivery of re-encrypted ciphertext and re-encryption key.

Achieve CCA Security. As discussed in Section 1.3, the biggest challenge is how to achieve CCA security while not jeopardizing the properties of attribute-based re-encryption, unidirectionality, and collusion resistance. In our construction, we use a technique, which is somewhat identical to the FO [36] conversion, to capture CCA security. Specifically, in the construction of ciphertext we leverage a TCR hash function to 'sign' the ciphertext's components as well as the description of LSSS; and meanwhile, we generate a 'verification key' to check the validity of the 'signature'. Note that the definition of the variables used later can be found in our scheme. In algorithm Enc, it can be seen that D

DOI: 10.1002/cpe

is the 'signature' and A_3 is the verification key such that the validity of ciphertext can be checked by $e(A_2, g_1) \overset{?}{=} e(g, A_3)$ and $e(A_3, H_4(A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), (M, \rho))) \overset{?}{=} e(g_1, D)$. In algorithm $ReEnc$, the proxy can first check the previous equations so as to guarantee the re-encryption to intake valid input. In addition, the well-formness of the components $\{(B_i, C_i) | 1 \leq i \leq l\}$ should be verified as they are the input of re-encryption as well. To capture this verification, we let the proxy check $e(\prod_{i \in I} B_i^{w_i}, g) \overset{?}{=} e(A_2, g^a) \cdot \prod_{i \in I} (e(C_i^{-1}, H_3(\rho(i))^{w_i}))$. After finishing the re-encryption, the proxy will output the re-encrypted ciphertext $(S, (M, \rho), A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), D, A_4, rk_4)$. For a legitimate delegatee, he or she is able to check the validity of the re-encrypted ciphertext as (M, ρ), A_1, A_3, $(B_1, C_1), \ldots, (B_l, C_l)$ are 'signed' in D and S is 'signed' in D', one of the components of rk_4 (which will be further elaborated later). Besides, A_4 is tightly related to the original ciphertext's components A_1 and A_3 in the sense that $A_3 \overset{?}{=} g_1^{H_1(m,\beta)}$ is able to tell whether A_4 is mutated or not.

As to the generation of re-encryption key $rk_{S \to (M', \rho')}$ from an attribute set S to a new access policy (M', ρ'), it can be seen that S and (M', ρ') are 'signed' in D', and the signature can be checked by A'_2. rk_1, rk_3, R_x are tightly related to rk_4 via δ, and rk_1 is bound with rk_2 with θ, where rk_4 is the encryption of δ under (M', ρ') with CCA security. Here if rk_1, rk_2, rk_3 and R_x are mutated by an adversary, the re-encryption then will yield an invalid result, which reveals no information of the underlying plaintext; on the other hand, if the description of S and (M', ρ'), and rk_4 are mutated, the proxy can tell by checking $e(A'_2, H_6(A'_1, A'_2, (B'_1, C'_1), \ldots, (B'_{l'}, C'_{l'}), S, (M', \rho'))) \overset{?}{=} e(g, D')$. Thus, the construction of re-encryption key precludes an adversary from constructing a new and valid re-encryption key either $rk_{S' \to (M', \rho')}$ or $rk_{S \to (M'', \rho'')}$ with knowledge of $rk_{S \to (M', \rho')}$.

Taking a close look at the algorithm $ReKeyGen$, we can see that the private key of the delegator is the only secret information required to be the input of the re-encryption key generation. Accordingly, our scheme is non-interactive in the generation of re-encryption key (which saves the bandwidth of communication) and unidirectional. Due to the subtle construction of rk_1 and rk_2 an adversary cannot compromise the entire private key of the delegator without knowledge of θ even if colluding with the corresponding delegatee. This captures collusion resistance. As to the single-hop property, it can be achieved as follows. Algorithm $ReEnc$ shows that A_3 (i.e., g_1^s) is a necessary component for re-encryption. Nevertheless, such a component is excluded in rk_4 such that rk_4 cannot be further re-encrypted. Thus, our scheme is single-hop.

4.2. Construction

The description of our new CP-ABPRE scheme with CCA security is as follows.

(1) $Setup(1^k, \mathcal{U})$. Given a security parameter k and $\mathcal{U}$, run $(p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow BSetup(1^k)$. Choose two random values $a, \alpha \in \mathbb{Z}_p^*$, a random generator $g_1 \in \mathbb{G}$, and the following TCR hash functions $H_1 : \{0,1\}^{2k} \to \mathbb{Z}_p^*$, $H_2 : \mathbb{G}_T \to \{0,1\}^{2k}$, $H_3 : \{0,1\}^* \to \mathbb{G}$, $H_4 : \{0,1\}^* \to \mathbb{G}$, $H_5 : \{0,1\}^k \to \mathbb{Z}_p^*$, $H_6 : \{0,1\}^* \to \mathbb{G}$. The public parameters are $param = (p, g, \mathbb{G}, \mathbb{G}_T, e, g_1, g^a, e(g,g)^\alpha, H_1, H_2, H_3, H_4, H_5, H_6)$, and the master secret key is $msk = g^\alpha$.

(2) $KeyGen(msk, S)$. Given msk and an attribute set S, choose $t \in_R \mathbb{Z}_p^*$, and set sk_S as

$$K = g^{a \cdot t} g^\alpha, L = g^t, \forall x \in S \ K_x = H_3(x)^t.$$

(3) $Enc((M, \rho), m)$. Taking an LSSS access structure (M, ρ) (M is an $l \times n$ matrix, and the function ρ associates rows of M to attributes) and a message $m \in \{0,1\}^k$ as input, the encryption algorithm works as follows.

   (a) Choose $\beta \in_R \{0,1\}^k$, set $s = H_1(m, \beta)$ and a random vector $v = (s, y_2, \ldots, y_n)$, where $y_2, \ldots, y_n \in_R \mathbb{Z}_p^*$.

   (b) For $i = 1$ to l, set $\lambda_i = vM_i$, where M_i is the vector corresponding to the ith row of M.

   (c) Choose $r_1, \ldots, r_l \in_R \mathbb{Z}_p^*$, set the original ciphertext as

$$A_1 = (m||\beta) \oplus H_2(e(g,g)^{\alpha \cdot s}), A_2 = g^s, A_3 = g_1^s, B_1 = (g^a)^{\lambda_1} H_3(\rho(1))^{-r_1}, C_1 = g^{r_1},$$
$$\ldots, B_l = (g^a)^{\lambda_l} H_3(\rho(l))^{-r_l}, C_l = g^{r_l}, D = H_4(A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), (M, \rho))^s,$$

DOI: 10.1002/cpe

and output $C_{(M,\rho)} = ((M, \rho), A_1, A_2, A_3, (B_1, C_1), \ldots, (B_l, C_l), D)$. Note $\{\rho(i) | 1 \leq i \leq l\}$ are the attributes used in the access structure (M, ρ). Like [22], we allow an attribute to be associated with multiple rows of matrix M, that is, the function ρ is not injective.

(4) $ReKeyGen(sk_S, S, (M', \rho'))$. Given a private key $sk_S = (K, L, \forall x \in S\ K_x)$, the attribute set S, and an LSSS access structure (M', ρ'), the re-encryption key is generated as follows, where M' is an $l' \times n'$ matrix, and the function ρ' associates rows of M' to attributes.

- The delegator generates the following encryption:

  (a) Choose $\beta', \delta \in_R \{0, 1\}^k$, set
  $s' = H_1(\delta, \beta')$ and a random vector $v' = (s', y'_2, \ldots, y'_{n'})$, where $y'_2, \ldots, y'_{n'} \in_R \mathbb{Z}_p^*$.
  (b) For $i = 1$ to l', set $\lambda'_i = v' M'_i$, where M'_i is the vector corresponding to the ith row of M'.
  (c) Choose $r'_1, \ldots, r'_{l'} \in_R \mathbb{Z}_p^*$, compute $A'_1 = (\delta || \beta') \oplus H_2(e(g, g)^{\alpha \cdot s'})$, $A'_2 = g^{s'}$, $B'_1 = (g^a)^{\lambda'_1} H_3(\rho'(1))^{-r'_1}$, $C'_1 = g^{r'_1}, \ldots, B'_{l'} = (g^a)^{\lambda'_{l'}} H_3(\rho'(l'))^{-r'_{l'}}$, $C'_{l'} = g^{r'_{l'}}$, $D' = H_6(A'_1, A'_2, (B'_1, C'_1), \ldots, (B'_{l'}, C'_{l'}), S, (M', \rho'))^{s'}$, and set $C_{(M',\rho')} = ((M', \rho'), A'_1, A'_2, (B'_1, C'_1), \ldots, (B'_{l'}, C'_{l'}), D')$.

- The delegator chooses $\theta \in_R \mathbb{Z}_p^*$, and sets $rk_1 = K^{H_5(\delta)} g_1^{\theta}, rk_2 = g^{\theta}, rk_3 = L^{H_5(\delta)}$, $\forall x \in S\ R_x = K_x^{H_5(\delta)}, rk_4 = C_{(M',\rho')}$, and outputs the re-encryption key $rk_{S \to (M',\rho')} = (S, rk_1, rk_2, rk_3, rk_4, \forall x \in S\ R_x)$.

(5) $ReEnc(rk_{S \to (M',\rho')}, C_{(M,\rho)})$. Parse the original ciphertext $C_{(M,\rho)}$ as $((M, \rho)$, A_1, A_2, A_3, (B_1, C_1), $\ldots$, (B_l, C_l), $D)$, and the re-encryption key $rk_{S \to (M',\rho')}$ as $(S, rk_1, rk_2, rk_3, rk_4, \forall x \in S\ R_x)$. Let $I \subseteq \{1, \ldots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$, $\{w_i \in \mathbb{Z}_p^*\}_{i \in I}$ be a set of constants such that $\sum_{i \in I} w_i \lambda_i = s$ if $\{\lambda_i\}$ are valid shares of any secret s according to M and $S \models (M, \rho)$. As stated in [22, 34], one can find the values w_i satisfying $\sum_{i \in I} w_i \lambda_i = s$ with knowledge of M and I.

  (a) Verify whether the re-encryption key $rk_{S \to (M',\rho')}$ contains valid S and (M', ρ') or not

  $$e\left(A'_2, H_6\left(A'_1, A'_2, \left(B'_1, C'_1\right), \ldots, \left(B'_{l'}, C'_{l'}\right), S, \left(M', \rho'\right)\right)\right) \overset{?}{=} e\left(g, D'\right).$$

  (b) Check the validity of the original ciphertext

  $$e(A_2, g_1) \overset{?}{=} e(g, A_3), e(A_3, H_4(A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), (M, \rho))) \overset{?}{=} e(g_1, D),$$
  $$e\left(\prod_{i \in I} B_i^{w_i}, g\right) \overset{?}{=} e(A_2, g^a) \prod_{i \in I} \left(e(C_i^{-1}, H_3(\rho(i))^{w_i})\right), S \overset{?}{\models} (M, \rho). \tag{1}$$

  If Equation (1) does not hold, output $\perp$. Otherwise, proceed.
  (c) Compute $A_4 = \frac{e(A_2, rk_1)/e(A_3, rk_2)}{(\prod_{i \in I}(e(B_i, rk_3) \cdot e(C_i, R_{\rho(i)}))^{w_i})}$, and output the re-encrypted ciphertext $C^R_{(M',\rho')} = (S, (M, \rho), A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), D, A_4, rk_4)$.

(6) $Dec(S, sk_S, C_{(M,\rho)})$. Parse the original ciphertext $C_{(M,\rho)}$ as $((M, \rho)$, A_1, A_2, A_3, (B_1, C_1), $\ldots$, (B_l, C_l), $D)$, and the private key sk_S as $(K, L, \forall x \in S\ K_x)$. Let $I \subseteq \{1, \ldots, l\}$ be defined as $I = \{i : \rho(i) \in S\}$, $\{w_i \in \mathbb{Z}_p^*\}_{i \in I}$ be a set of constants such that $\sum_{i \in I} w_i \lambda_i = s$.

  (1) Verify Equation (1). If Equation (1) does not hold, output $\perp$. Otherwise, proceed.
  (2) Compute $Z = e(A_2, K)/(\prod_{i \in I}(e(B_i, L) \cdot e(C_i, K_{\rho(i)}))^{w_i})$ and $m || \beta = H_2(Z) \oplus A_1$, output m if
  $A_3 = g_1^{H_1(m,\beta)}$, and output $\perp$ otherwise.

(7) $Dec_R(S', sk_{S'}, C^R_{(M',\rho')})$. Parse the re-encrypted ciphertext $C^R_{(M',\rho')}$ as $(S$, (M, ρ), A_1, A_3, $(B_1, C_1), \ldots, (B_l, C_l)$, D, A_4, $rk_4)$, and the private key $sk_{S'}$ as $(K', L', \forall x \in S'\ K'_x)$.

DOI: 10.1002/cpe

(a) Recover $\delta||\beta'$ as follows. Let $I' \subseteq \{1, \ldots, l'\}$ be defined as $I' = \{i : \rho'(i) \in S'\}$, $\{w'_i \in \mathbb{Z}_p^*\}_{i \in I'}$ be a set of constants such that $\sum_{i \in I'} w'_i \lambda'_i = s'$ if $\{\lambda'_i\}$ are valid shares of any secret s' according to M' and $S' \models (M', \rho')$.

(i) Verify

$$e(A'_2, H_6(A'_1, A'_2, (B'_1, C'_1), \ldots, (B'_{l'}, C'_{l'}), S, (M', \rho'))) \overset{?}{=} e(g, D'), S' \overset{?}{\models} (M', \rho'). \qquad (2)$$

If Equation (2) does not hold, output $\perp$. Otherwise, proceed.

(ii) Compute $Z' = e(A'_2, K')/\left(\prod_{i \in I'} \left(e\left(B'_i, L'\right) \cdot e\left(C'_i, K'_{\rho'(i)}\right)\right)^{w'_i}\right)$ and $\delta||\beta' = H_2(Z') \oplus A'_1$, proceed if $A'_2 = g^{H_1(\delta, \beta')}$, and output $\perp$ otherwise.

(b) Compute $m||\beta = H_2(A_4^{\frac{1}{H_5(\delta)}}) \oplus A_1$, output m if $A_3 = g_1^{H_1(m,\beta)}$, $D = H_4(A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), (M, \rho))^{H_1(m,\beta)}$ and $S \models (M, \rho)$, and output $\perp$ otherwise.

- *Correctness for original ciphertext.*

$$\begin{aligned} Z &= e(A_2, K)/\left(\prod_{i \in I} \left(e\left(B_i, L\right) \cdot e\left(C_i, K_{\rho(i)}\right)\right)^{w_i}\right) \\ &= \frac{e(g^s, g^{a \cdot t} \cdot g^{\alpha})}{\left(\prod_{i \in I} (e(g^{a \cdot \lambda_i} \cdot H_3(\rho(i))^{-r_i}, g^t) \cdot e(g^{r_i}, H_3(\rho(i))^t))^{w_i}\right)} \\ &= \frac{e(g^s, g^{a \cdot t} \cdot g^{\alpha})}{e(g, g^{a \cdot t})^{\sum_{i \in I} \lambda_i \cdot w_i}} = e(g^s, g^{\alpha}), \end{aligned}$$

we have $H_2(Z) \oplus A_1 = H_2(e(g^s, g^{\alpha})) \oplus (m||\beta) \oplus H_2(e(g, g)^{\alpha \cdot s}) = m||\beta$.

- *Correctness for re-encrypted ciphertext.*

$$\begin{aligned} A_4 &= \frac{e(A_2, rk_1)/e(A_3, rk_2)}{\left(\prod_{i \in I} (e(B_i, rk_3) \cdot e(C_i, R_{\rho(i)}))^{w_i}\right)} \\ &= \frac{e(g^s, (g^{a \cdot t} \cdot g^{\alpha})^{H_5(\delta)} \cdot g_1^{\theta})/e(g_1^s, g^{\theta})}{\prod_{i \in I} (e((g^a)^{\lambda_i} \cdot H_3(\rho(i))^{-r_i}, (g^t)^{H_5(\delta)}) \cdot e(g^{r_i}, H_3(\rho(i))^{t \cdot H_5(\delta)}))^{w_i}} \\ &= \frac{e\left(g^s, g^{\alpha \cdot H_5(\delta)}\right) \cdot e\left(g^s, g^{a \cdot t \cdot H_5(\delta)}\right)}{e\left(g, g^{a \cdot t \cdot H_5(\delta)}\right)^{\sum_{i \in I} \lambda_i \cdot w_i}} = e\left(g^s, g^{\alpha \cdot H_5(\delta)}\right), \end{aligned}$$

$$\begin{aligned} Z' &= e(A'_2, K')/\left(\prod_{i \in I'} \left(e\left(B'_i, L'\right) \cdot e\left(C'_i, K'_{\rho'(i)}\right)\right)^{w'_i}\right) \\ &= \frac{e(g^{s'}, g^{a \cdot t} \cdot g^{\alpha})}{\left(\prod_{i \in I'} (e(g^{a \cdot \lambda'_i} \cdot H_3(\rho'(i))^{-r'_i}, g^t) \cdot e(g^{r'_i}, H_3(\rho'(i))^t))^{w'_i}\right)} \\ &= \frac{e(g^{s'}, g^{a \cdot t} \cdot g^{\alpha})}{e(g, g^{a \cdot t})^{\sum_{i \in I'} \lambda'_i \cdot w'_i}} = e(g^{s'}, g^{\alpha}), \end{aligned}$$

we have $H_2(Z') \oplus A'_1 = H_2(e(g^{s'}, g^{\alpha})) \oplus (\delta||\beta') \oplus H_2(e(g, g)^{\alpha \cdot s'}) = \delta||\beta'$, and $H_2(A_4^{\frac{1}{H_5(\delta)}}) \oplus A_1 = H_2(e(g, g)^{\alpha s H_5(\delta)})^{\frac{1}{H_5(\delta)}} \oplus (m||\beta) \oplus H_2(e(g, g)^{\alpha s}) = m||\beta$.

4.3. Security analysis

Before giving the formal security analysis, we first give some intuition as to why our CP-ABPRE scheme is secure against CCA. For the security of original ciphertext, let $C^*_{(M^*, \rho^*)} = ((M^*, \rho^*), A_1^*, A_2^*, A_3^*, (B_1^*, C_1^*), \ldots, (B_l^*, C_l^*), D^*)$ be the challenge original ciphertext of m_b. Suppose an adversary $\mathcal{A}$ following the constraints given in Definition 4 will try to take additional advantage in guessing the value of the bit b by leveraging the responses of $\mathcal{O}_{re}$ and $\mathcal{O}_{d2}$. Specifically, $\mathcal{A}$ might mutate $C^*_{(M^*,\rho^*)}$, and issue the resulting ciphertext to $\mathcal{O}_{re}$ and $\mathcal{O}_{d2}$. From Equation (1), the mutation can be noticeable with non-negligible probability. This is so because A_1^*, A_3^*, (B_1^*, C_1^*),..., (B_l^*, C_l^*) are bound by D as well as the description of (M^*, ρ^*). Note that D can be viewed as a signature for the previous components. Besides, the integrity of A_2^* is bound by A_3^*. If the challenge ciphertext is mutated, Equation (1) will not hold. Therefore, no extra advantage in guessing b leaks to $\mathcal{A}$.

For the security of re-encrypted ciphertext, let $C^{R*}_{(M^*,\rho^*)} = (S, (M,\rho), A_1^*, A_3^*, (B_1^*, C_1^*), \ldots, (B_l^*, C_l^*), D^*, A_4^*, rk_4^*)$ be the challenge re-encrypted ciphertext of m_b. Following Definition 5, $\mathcal{A}$ will try to obtain additional advantage in winning the game with the help of $\mathcal{O}_{d1}$. Before proceeding, we show that the re-encrypted ciphertext cannot be re-encrypted, that is, given $\mathcal{O}_{re}$, $\mathcal{A}$ cannot achieve extra advantage. It is not difficult to see that rk_4^* cannot be re-encrypted without A_3' (i.e., $g_1^{s'}$), which is a crucial component for re-encryption. Furthermore, A_2^* that is needed in re-encryption and the verification in Equation (1) is excluded in the re-encrypted ciphertext as well.

Accordingly, the re-encryption query for any re-encrypted ciphertext will be rejected.

Given $C^{R*}_{(M^*,\rho^*)}$ $\mathcal{A}$ cannot mutate the ciphertext and issue the resulting ciphertext to $\mathcal{O}_{d1}$ such that the oracle outputs a valid decryption result without any rejection. The reason is that A_1^*, A_3^*, (B_1^*, C_1^*), ..., (B_l^*, C_l^*) are still bound by D^* as well as the description of (M, ρ); meanwhile, S and the description of (M^*, ρ^*) are bound by D', a component of rk_4^*. Note that rk_4^* is secure against CCA. D' can be regarded as a signature for all the components contained in rk_4^* (except D' itself) and S, and A_2' can be seen as the verification key. Here the only consideration left is the integrity of A_4^*. We state that if A_4^* is mutated by $\mathcal{A}$, the challenger can tell the change with non-negligible probability. Please refer to our formal security proof for the details. Hence, $\mathcal{A}$ cannot acquire additional advantage in winning the game by using $\mathcal{O}_{d1}$.

Therefore, we have the following theorem.

Theorem 2
Suppose the decisional q-parallel BDHE assumption holds in $(\mathbb{G}, \mathbb{G}_T)$, and H_1, H_2, H_3, H_4, H_5, H_6 are the TCR hash functions, our CP-ABPRE scheme is IND-sAS-CCA secure in the random oracle model.

Proof
IND-sAS-CCA-Or Security.

Suppose there exists an adversary $\mathcal{A}$ who can break the IND-sAS-CCA-Or security of our scheme. We then construct a reduction algorithm $\mathcal{C}$ to decide whether $T = e(g,g)^{a^{q+1}\cdot s}$ or $T \in_R \mathbb{G}_T$. $\mathcal{C}$ plays the IND-sAS-CCA-Or game with $\mathcal{A}$ as follows.

$\mathcal{C}$ takes in $(p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow BSetup(1^k)$ and a q-parallel BDHE instance y and $T \in \mathbb{G}_T$, where T is either equal to $e(g,g)^{a^{q+1}\cdot s}$ or equal to $T' \in_R \mathbb{G}_T$.

(1) *Initialization.* $\mathcal{A}$ outputs (M^*, ρ^*) to $\mathcal{C}$, where M^* is an $l^* \times n^*$ matrix, and $l^*, n^* \leq q$.

(2) *Setup.* $\mathcal{C}$ chooses $\alpha', \gamma \in_R \mathbb{Z}_p^*$ and sets $g_1 = g^\gamma$, $e(g,g)^\alpha = e(g^a, g^{a^q}) \cdot e(g, g^{\alpha'})$. It implicitly sets $\alpha = \alpha' + a^{q+1}$ (which cannot be computed by $\mathcal{C}$). Then $\mathcal{C}$ chooses the TCR hash functions as in the real scheme, and sends the public parameters $param = (p, g, \mathbb{G}, \mathbb{G}_T, e, g_1, g^a, e(g,g)^\alpha, H_1, H_2, H_3, H_4, H_5, H_6)$ to $\mathcal{A}$. From the point of view of $\mathcal{A}$, the public parameters are identical to those of the real scheme. At any time, $\mathcal{A}$ can adaptively query the random oracles H_j $(j \in \{1, \ldots, 6\})$, which are controlled by $\mathcal{C}$. $\mathcal{C}$ maintains the lists H_j^{List} $(j \in \{1, \ldots, 6\})$, which are initially empty, and answers the queries to the random oracles as follows.

(a) H_1: on receipt of an H_1 query on (m, β), if there is a tuple (m, β, s) in H_1^{List}, $\mathcal{C}$ forwards the predefined value s to $\mathcal{A}$, where $s \in \mathbb{Z}_p^*$. Otherwise, $\mathcal{C}$ sets $H_1(m, \beta) = s$, responds s to $\mathcal{A}$ and adds the tuple (m, β, s) to H_1^{List}, where $s \in_R \mathbb{Z}_p^*$.

(b) H_2: on receipt of an H_2 query on $R \in \mathbb{G}_T$, if there is a tuple (R, δ_1) in H_2^{List}, $\mathcal{C}$ forwards the predefined value δ_1 to $\mathcal{A}$, where $\delta_1 \in \{0,1\}^{2k}$. Otherwise, $\mathcal{C}$ sets $H_2(R) = \delta_1$, responds δ_1 to $\mathcal{A}$ and adds the tuple (R, δ_1) to H_2^{List}, where $\delta_1 \in_R \{0,1\}^{2k}$.

(c) H_3: on receipt of an H_3 query on $x \in \mathcal{U}$, if there is a tuple $(x, z_x, \delta_{2,x})$ in H_3^{List}, $\mathcal{C}$ forwards the predefined value $\delta_{2,x}$ to $\mathcal{A}$, where $z_x \in \mathbb{Z}_p^*$, $\delta_{2,x} \in \mathbb{G}$. Otherwise, $\mathcal{C}$ constructs $\delta_{2,x}$ as follows. Let X denote the set of indices i such that $\rho^*(i) = x$, where $1 \leq i \leq l^*$. Namely, X contains the indices of rows of matrix M^* that corresponds to the same attribute x. $\mathcal{C}$ chooses $z_x \in_R \mathbb{Z}_p^*$ and sets

$$\delta_{2,x} = g^{z_x} \cdot \prod_{i \in X} g^{a \cdot M_{i,1}^*/b_i + a^2 \cdot M_{i,2}^*/b_i + \cdots + a^{n^*} \cdot M_{i,n^*}^*/b_i}.$$

If $X = \emptyset$, $\mathcal{C}$ sets $\delta_{2,x} = g^{z_x}$. $\mathcal{C}$ responds $\delta_{2,x}$ to $\mathcal{A}$ and adds the tuple $(x, z_x, \delta_{2,x})$ to H_3^{List}.

(d) H_4: on receipt of an H_4 query on $(A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), (M, \rho))$, if there is a tuple $(A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), (M, \rho), \xi_1, \delta_3)$ in H_4^{List}, $\mathcal{C}$ forwards the predefined value δ_3 to $\mathcal{A}$, where $\xi_1 \in \mathbb{Z}_p^*$, $\delta_3 \in \mathbb{G}$. Otherwise, $\mathcal{C}$ sets $\delta_3 = g^{\xi_1}$, responds δ_3 to $\mathcal{A}$ and adds the tuple $(A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), (M, \rho), \xi_1, \delta_3)$ to H_4^{List}, where $\xi_1 \in_R \mathbb{Z}_p^*$.

(e) H_5: on receipt of an H_5 query on $\delta \in \{0,1\}^k$, if there is a tuple (δ, ξ_2) in H_5^{List}, $\mathcal{C}$ forwards the predefined value ξ_2 to $\mathcal{A}$, where $\xi_2 \in \mathbb{Z}_p^*$. Otherwise, $\mathcal{C}$ sets $H_5(\delta) = \xi_2$, responds ξ_2 to $\mathcal{A}$ and adds the tuple (δ, ξ_2) to H_5^{List}, where $\xi_2 \in_R \mathbb{Z}_p^*$.

(f) H_6: on receipt of an H_6 query on $(A_1', A_2', (B_1', C_1'), \ldots, (B_{l'}', C_{l'}'), S, (M', \rho'))$, if there is a tuple $(A_1', A_2', (B_1', C_1'), \ldots, (B_{l'}', C_{l'}'), S, (M', \rho'), \xi_3, \delta_4)$ in H_6^{List}, $\mathcal{C}$ forwards the predefined value δ_4 to $\mathcal{A}$, where $\xi_3 \in \mathbb{Z}_p^*$, $\delta_4 \in \mathbb{G}$. Otherwise, $\mathcal{C}$ sets $\delta_4 = g^{\xi_3}$, responds δ_4 to $\mathcal{A}$ and adds the tuple $(A_1', A_2', (B_1', C_1'), \ldots, (B_{l'}', C_{l'}'), S, (M', \rho'), \xi_3, \delta_4)$ to H_6^{List}, where $\xi_3 \in_R \mathbb{Z}_p^*$.

In addition, $\mathcal{C}$ also maintains the following lists, which are initially empty.

(a) SK^{List} records the tuples (S, sk_S), which are the responses of the queries to $\mathcal{O}_{sk}(S)$.

(b) RK^{List} records the tuples $(S, (M', \rho'), \delta, \beta', rk_{S \to (M', \rho')}, tag_1, tag_2, tag_3)$, which are the responses of the queries to $\mathcal{O}_{rk}(S, (M', \rho'))$, where tag_1, tag_2, tag_3 denote that the re-encryption key is randomly chosen, generated in $\mathcal{O}_{re}$ or in $\mathcal{O}_{rk}$, respectively.

(c) RE^{List} records the tuples $(S, (M', \rho'), C^R_{(M', \rho')}, tag_1, tag_2, tag_3)$, which are the responses of the queries to $\mathcal{O}_{re}(S, (M', \rho'), C_{(M, \rho)})$, where tag_1, tag_2, tag_3 denote that the re-encrypted ciphertext is generated under a valid re-encryption key, under a randomly chosen re-encryption key or generated without any re-encryption key.

(3) *Phase 1.* $\mathcal{A}$ issues a series of queries to which $\mathcal{C}$ responds as follows.

(a) *Private key extraction oracle* $\mathcal{O}_{sk}(S)$: $\mathcal{C}$ constructs the private key sk_S as follows. If $S \models (M^*, \rho^*)$, $\mathcal{C}$ then outputs $\perp$. Otherwise, that is $S \not\models (M^*, \rho^*)$, $\mathcal{C}$ chooses $r \in_R \mathbb{Z}_p^*$, $w = (w_1, \ldots, w_{n^*}) \in_R \mathbb{Z}_p^{*n^*}$ such that $w_1 = -1$ and $\forall i\ \rho^*(i) \in S$ we have $w \cdot M_i^* = 0$. Note that this vector w must exist by the convention of an LSSS, as stated in [22]. $\mathcal{C}$ then sets $L = g^r \cdot \prod_{i=1,\ldots,n^*} g^{a^{q+1-i} \cdot w_i} = g^t$. Here t is implicitly defined as $t = r + w_1 \cdot a^q + \cdots + w_{n^*} \cdot a^{q-n^*+1}$. $\mathcal{C}$ further constructs K as $K = g^{\alpha'} \cdot g^{a \cdot r} \cdot \prod_{i=2,\ldots,n^*} g^{a^{q+2-i} \cdot w_i}$. One can verify that K is valid

$$K = g^{\alpha'} \cdot g^{a \cdot r} \cdot \prod_{i=2,\ldots,n^*} g^{a^{q+2-i} \cdot w_i} = g^{\alpha'} \cdot g^{a^{q+1}} \cdot g^{-a^{q+1}} \cdot g^{a \cdot r} \cdot \prod_{i=2,\ldots,n^*} g^{a^{q+2-i} \cdot w_i}$$

$$= g^{\alpha} \cdot (g^r \cdot \prod_{i=1,\ldots,n^*} g^{a^{q+1-i} \cdot w_i})^a = g^{\alpha} \cdot L^a = g^{\alpha} \cdot g^{a \cdot t}.$$

If $x \in S$ but $\rho^*(i) \neq x$ for any $i \in \{1, \ldots, l^*\}$, $\mathcal{C}$ then sets $K_x = L^{z_x}$. It is easy to see that $K_x = L^{z_x} = (g^t)^{z_x} = \delta_{2,x}^t = H_3(x)^t$. Otherwise, $\mathcal{C}$ constructs K_x as $K_x = L^{z_x} \cdot \prod_{i \in X} \prod_{j=1,\ldots,n^*} (g^{(a^j/b_i) \cdot r} \cdot \prod_{k=1,\ldots,n^*, k \neq j} (g^{a^{q+1+j-k}/b_i})^{w_k})^{M_{i,j}^*}$. It can be seen that K_x is valid

$$
\begin{aligned}
K_x &= L^{z_x} \cdot \prod_{i \in X} \prod_{j=1,\ldots,n^*} \left(g^{(a^j/b_i)\cdot r} \cdot \prod_{k=1,\ldots,n^*, k \neq j} \left(g^{a^{q+1+j-k}/b_i} \right)^{w_k} \right)^{M^*_{i,j}} \\
&= L^{z_x} \cdot \prod_{i \in X} \prod_{j=1,\ldots,n^*} \left(g^{(a^j/b_i)\cdot r} \cdot \prod_{k=1,\ldots,n^*, k \neq j} \left(g^{a^{q+1+j-k}/b_i} \right)^{w_k} \right)^{M^*_{i,j}} \\
&\quad \cdot \prod_{i \in X} \prod_{j=1,\ldots,n^*} \left(g^{a^{q+1}/b_i} \right)^{w_j \cdot M^*_{i,j}} \\
&= \left(g^r \cdot \prod_{i=1,\ldots,n^*} g^{a^{q+1-i}\cdot w_i} \right)^{z_x} \cdot \prod_{i \in X} \prod_{j=1,\ldots,n^*} \\
&\quad \left(g^{(a^j/b_i)\cdot r} \cdot \prod_{k=1,\ldots,n^*} \left(g^{a^{q+1+j-k}/b_i} \right)^{w_k} \right)^{M^*_{i,j}} \\
&= \left(g^{z_x} \cdot \prod_{i \in X} g^{a \cdot M^*_{i,1}/b_i + a^2 \cdot M^*_{i,2}/b_i + \cdots + a^{n^*} \cdot M^*_{i,n^*}/b_i} \right)^{\left(r + w_1 \cdot a^q + \cdots + w_{n^*} \cdot a^{q-n^*+1}\right)} \\
&= \delta_{2,x}^{\left(r + w_1 \cdot a^q + \cdots + w_{n^*} \cdot a^{q-n^*+1}\right)} = \delta^t_{2,x} = H_3(x)^t,
\end{aligned}
$$

where X is the set of all i such that $\rho^*(i) = x$. Recall that if $S \not\models (M^*, \rho^*)$, we then have $w \cdot M^*_i = 0$. Thus, we have $\prod_{i \in X} \prod_{j=1,\ldots,n^*} (g^{a^{q+1}/b_i})^{w_j \cdot M^*_{i,j}} = g^{a^{q+1} \cdot (\sum_{i \in X} \sum_{j=1,\ldots,n^*} w_j \cdot M^*_{i,j}/b_i)} = g^0 = 1$. Finally, $\mathcal{C}$ adds the tuple (S, sk_S) to SK^{List} and returns sk_S to $\mathcal{A}$.

(b) *Re-encryption key extraction oracle* $\mathcal{O}_{rk}(S, (M', \rho'))$: if $(S$, (M', ρ'), δ, β', $rk_{S \to (M',\rho')}$, $*, 0, 1) \in RK^{List}$, $\mathcal{C}$ returns $rk_{S \to (M',\rho')}$ to $\mathcal{A}$. Otherwise, $\mathcal{C}$ proceeds.

- If $S \models (M^*, \rho^*) \wedge (S', sk_{S'}) \in SK^{List}$ (for any $S' \models (M', \rho')$), $\mathcal{C}$ outputs $\perp$.
- If $S \models (M^*, \rho^*) \wedge (S', sk_{S'}) \notin SK^{List}$ (for any $S' \models (M', \rho')$), $\mathcal{C}$ checks whether $(S$, (M', ρ'), δ, β', $rk_{S \to (M',\rho')}$, $1, 1, 0) \in RK^{List}$. If yes, $\mathcal{C}$ returns $rk_{S \to (M',\rho')}$ to $\mathcal{A}$ and resets $tag_2 = 0, tag_3 = 1$. Otherwise, $\mathcal{C}$ first chooses $\theta, \sigma \in_R \mathbb{Z}^*_p$, $\beta', \delta \in_R \{0,1\}^k$, $\bar{K} \in_R \mathbb{G}$. $\mathcal{C}$ then sets $rk_1 = \bar{K} \cdot g_1^{\theta}$, $rk_2 = g^{\theta}$, $rk_3 = g^{\sigma}$, $R_x = \delta^{\sigma}_{2,x}$,

  and constructs rk_4 as in the real scheme, where $\delta_{2,x}$ is the output of issuing x to H_3, $x \in S$.

  Finally, $\mathcal{C}$ returns $rk_{S \to (M',\rho')} = (S, rk_1, rk_2, rk_3, rk_4, R_x)$ to $\mathcal{A}$, and adds $(S, (M', \rho'), \delta, \beta', rk_{S \to (M',\rho')}, 1, 0, 1)$ to RK^{List}.
- Otherwise, if $(S$, (M', ρ'), δ, β', $rk_{S \to (M',\rho')}$, $0, 1, 0) \in RK^{List}$, $\mathcal{C}$ returns $rk_{S \to (M',\rho')}$ to $\mathcal{A}$, and resets $tag_2 = 0, tag_3 = 1$. Otherwise, $\mathcal{C}$ first constructs the private key sk_S as in step (a). $\mathcal{C}$ further generates $rk_{S \to (M',\rho')}$ as in the real scheme, returns the re-encryption key to $\mathcal{A}$ and adds $(S$, (M', ρ'), δ, β', $rk_{S \to (M',\rho')}$, $0, 0, 1)$ to RK^{List}.

(c) *Re-encryption oracle* $\mathcal{O}_{re}(S, (M', \rho'), C_{(M,\rho)})$: $\mathcal{C}$ verifies whether Equation (1) holds. If not (i.e., indicating that either the ciphertext $C_{(M,\rho)}$ is invalid or $S \not\models (M, \rho)$), $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ proceeds.

- If $S \models (M^*, \rho^*) \wedge (S', sk_{S'}) \in SK^{List}$ (for any $S' \models (M', \rho')$) does not hold,

  (i) If $S \models (M^*, \rho^*) \wedge (S', sk_{S'}) \notin SK^{List}$, $\mathcal{C}$ first constructs the re-encryption key as in the second case of step (b), further re-encrypts $C_{(M,\rho)}$ to $\mathcal{A}$, and finally adds $(S$, (M', ρ'), δ, β', $rk_{S \to (M',\rho')}$, $1, 1, 0)$, $(S, (M', \rho'), C^R_{(M',\rho')}, 0, 1, 0)$ to RK^{List}, RE^{List}, respectively.

(ii) Otherwise, $\mathcal{C}$ first constructs the re-encryption key as in the third case of step (b), further re-encrypts $C_{(M,\rho)}$ to $\mathcal{A}$, and finally adds $(S, (M', \rho'), \delta, \beta', rk_{S\rightarrow(M',\rho')}, 0, 1, 0)$, $(S, (M', \rho'), C^R_{(M',\rho')}, 1, 0, 0)$ to RK^{List}, RE^{List}, respectively.

- Otherwise, if $S \models (M^*, \rho^*) \wedge (S', sk_{S'}) \in SK^{List}$ (for any $S' \models (M', \rho')$) and $C_{(M,\rho)}$ is the challenge ciphertext, $\mathcal{C}$ outputs $\perp$. Else, $\mathcal{C}$ checks whether $(m, \beta, s) \in H_1^{List}$ such that $A_3 = g_1^s$. If no such tuple exists, $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ checks whether $(S, (M', \rho'), \delta, \beta', \perp, \perp, 1, \perp) \in RK^{List}$, where $S \models (M^*, \rho^*)$. If no, $\mathcal{C}$ chooses $\beta', \delta \in_R \{0, 1\}^k$, generates $rk_4 = C_{(M',\rho')}$ (to hide δ and β') as in the real scheme, and constructs $A_4 = (e(g^a, g^{a^q}) \cdot e(g, g^{\alpha'}))^{s\cdot\xi_2}$, where $\xi_2 = H_5(\delta)$. Finally, $\mathcal{C}$ returns $C^R_{(M',\rho')} = (S, (M, \rho), A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), D, A_4, rk_4)$ to $\mathcal{A}$, and adds $(S, (M', \rho'), \delta, \beta', \perp, \perp, 1, \perp)$, $(S, (M', \rho'), C^R_{(M',\rho')}, 0, 0, 1)$ to RK^{List}, RE^{List}, respectively.

(d) *Original ciphertext decryption oracle* $\mathcal{O}_{d2}(S, C_{(M,\rho)})$: $\mathcal{C}$ verifies whether Equation (1) holds. If not (i.e., indicating either the ciphertext is invalid or $S \not\models (M, \rho)$), $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ proceeds.

- If $(S, sk_S) \in SK^{List}$ (for any $S \models (M, \rho)$), $\mathcal{C}$ recovers m as in the real scheme by using sk_S.
- Otherwise, if $C_{(M,\rho)}$ is the challenge ciphertext, $\mathcal{C}$ outputs $\perp$. Else, $\mathcal{C}$ checks whether $(m, \beta, s) \in H_1^{List}$ and $(R, \delta_1) \in H_2^{List}$ such that $A_3 = g_1^s$, $A_1 = (m||\beta) \oplus \delta_1$ and $R = e(g, g)^{\alpha\cdot s}$. $\mathcal{C}$ outputs $\perp$ if no such tuples exist, and outputs m otherwise.

(e) *Re-encrypted ciphertext decryption oracle* $\mathcal{O}_{d1}(S', C^R_{(M',\rho')})$: $\mathcal{C}$ first checks whether there are tuples (δ, β', s') and (m, β, s) in H_1^{List} such that $A'_2 = g^{s'}$ and $A_3 = g_1^s$. If not, $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ verifies whether Equation (2) holds. If not (i.e., indicating either rk_4 is invalid or $S' \not\models (M', \rho')$), $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ proceeds.

- If $C^R_{(M',\rho')}$ is a derivative of the challenge ciphertext, $\mathcal{C}$ outputs $\perp$.
- If $(S, (M', \rho'), \delta, \beta', rk_{S\rightarrow(M',\rho')}, 1, 0, 1) \in RK^{List} \vee (S, (M', \rho'), C^R_{(M',\rho')}, 0, 1, 0) \in RE^{List}$, $\mathcal{C}$ checks

$$e\left(\prod_{i\in I'} B_i'^{w_i'}, g\right) \stackrel{?}{=} e\left(A_2', g^a\right) \cdot \prod_{i\in I'} \left(e\left(C_i'^{-1}, H_3(\rho'(i))^{w_i'}\right)\right), \tag{3}$$

where with knowledge of M' and I' ($I' \subseteq \{1, \ldots, l'\}$ and $I' = \{i : \rho'(i) \in S'\}$), $\mathcal{C}$ can find $\{w_i' \in \mathbb{Z}_p^*\}_{i\in I'}$ such that $\sum_{i\in I'} w_i' \cdot \lambda_i' = s'$. If Equation (3) does not hold, $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ reconstructs $A_2 = g^s$ with knowledge of s and then verifies Equation (1). If the equation does not hold, $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ recovers the random re-encryption key $rk_{S\rightarrow(M',\rho')} = (S, rk_1, rk_2, rk_3, rk_4, R_x)$ from RK^{List}, and checks the validity of A_4 as $A_4 \stackrel{?}{=} \frac{e(A_2, rk_1)/e(A_3, rk_2)}{(\prod_{i\in I}(e(B_i, rk_3)\cdot e(C_i, R_{\rho(i)}))^{w_i})}$, where I and w_i are already defined in algorithm $ReEnc$. If the previous equation does not hold, $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ checks whether $(R, \delta_1) \in H_2^{List}$ such that $A_1 = (m||\beta) \oplus \delta_1$ and $R = e(g, g)^{\alpha\cdot s}$. If no such tuple exists, $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ returns m to $\mathcal{A}$. Here it is worth mentioning that $\mathcal{C}$ can tell the derivatives of the challenge ciphertext via the previous verification approach.
- Otherwise,

(i) If $(S', sk_{S'}) \in SK^{List}$, $\mathcal{C}$ recovers m as in the real scheme by using $sk_{S'}$.

(ii) Otherwise, $\mathcal{C}$ checks whether Equation (3) holds. If not, $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ checks whether $(R, \delta_1) \in H_2^{List}$ such that $A_1 = (m||\beta) \oplus \delta_1$, $R = e(g, g)^{\alpha\cdot s}$, verifies whether $A_4 = e(g,g)^{\alpha \cdot s \cdot \xi_2}$ and $D = H_4(A_1, A_3, (B_1, C_1), \ldots, (B_l, C_l), (M, \rho))^s$ hold, where $\xi_2 = H_5(\delta)$. If no such tuple exists and the equations do not hold, $\mathcal{C}$ outputs $\perp$. Otherwise, $\mathcal{C}$ outputs m.

(4) *Challenge.* $\mathcal{A}$ outputs m_0, m_1 to $\mathcal{C}$. $\mathcal{C}$ chooses $b \in_R \{0,1\}$ and responds as follows.

(a) For each row i of M^*, set $x^* = \rho^*(i)$, and issue an H_3 query on x^* to obtain the tuple $(x^*, z_{x^*}, \delta_{2,x^*})$. Like [22] (the challenger is able to choose the secret splitting), choose $y'_2, \ldots, y'_{n^*}$ and share the secret using the vector $v = (s, s \cdot a + y'_2, s \cdot a^2 + y'_3, \ldots, s \cdot a^{n-1} + y'_{n^*}) \in \mathbb{Z}_p^{n^*}$. Choose $r'_1, \ldots, r'_{l^*} \in_R \mathbb{Z}_p^*$, for all $i \in \{1, \ldots, l^*\}$, denote R_i as the set of all $i \neq k$ such that $\rho^*(i) = \rho^*(k)$. Set

$$\begin{aligned} B_i^* &= \delta_{2,x^*}^{-r'_i} \cdot \left(\prod_{j=2,\ldots,n^*} g^{a \cdot M^*_{i,j} \cdot y'_j} \right) \cdot g^{b_i \cdot s \cdot (-z_{x^*})} \cdot \\ &\quad \left(\prod_{k \in R_i} \prod_{j=1,\ldots,n^*} \left(g^{a^j \cdot s \cdot (b_i/b_k)} \right)^{M^*_{k,j}} \right)^{-1}, \\ C_i^* &= g^{r'_i + s \cdot b_i}. \end{aligned}$$

(b) Choose $\beta^* \in_R \{0,1\}^k$, $A_1^* \in_R \{0,1\}^{2k}$, implicitly define $H_2(T \cdot e(g^s, g^{\alpha'})) = A_1^* \oplus (m_b || \beta^*)$, and set $A_2^* = g^s$, $A_3^* = (g^s)^\gamma$.

(c) Issue an H_4 query on $(A_1^*, A_3^*, (B_1^*, C_1^*), \ldots, (B_{l^*}^*, C_{l^*}^*), (M^*, \rho^*))$ to obtain the tuple $(A_1^*, A_3^*, (B_1^*, C_1^*), \ldots, (B_{l^*}^*, C_{l^*}^*), (M^*, \rho^*), \xi_1^*, \delta_3^*)$, and define $D^* = (g^s)^{\xi_1^*}$.

(d) Output the challenge original ciphertext $C^*_{(M^*,\rho^*)} = ((M^*, \rho^*), A_1^*, A_2^*, A_3^*, (B_1^*, C_1^*), \ldots, (B_{l^*}^*, C_{l^*}^*), D^*)$ to $\mathcal{A}$.

If $T = e(g,g)^{a^{q+1} \cdot s}$, $C^*_{(M^*,\rho^*)}$ is a valid ciphertext. Implicitly letting $H_1(m_b, \beta^*) = s$ and $r_i = r'_i + s \cdot b_i$, one can verify that

$$\begin{aligned} A_1^* &= A_1^* \oplus (m_b || \beta^*) \oplus (m_b || \beta^*) \\ &= H_2\left(T \cdot e\left(g^s, g^{\alpha'}\right)\right) \oplus (m_b || \beta^*) = H_2\left(e(g,g)^{\alpha \cdot s}\right) \oplus (m_b || \beta^*), \\ A_2^* &= g^s, A_3^* = (g^s)^\gamma = (g^\gamma)^s = g_1^s, \\ D^* &= (g^s)^{\xi_1^*} = \left(g^{\xi_1^*}\right)^s = H_4\left(A_1^*, A_3^*, (B_1^*, C_1^*), \ldots, (B_{l^*}^*, C_{l^*}^*), (M^*, \rho^*)\right)^s, \\ B_i^* &= \delta_{2,x^*}^{-r'_i} \left(\prod_{j=2,\ldots,n^*} (g^a)^{M^*_{i,j} \cdot y'_j} \right) \left(g^{b_i \cdot s}\right)^{-z_{x^*}} \left(\prod_{k \in R_i} \prod_{j=1,\ldots,n^*} \left(g^{a^j \cdot s \cdot (b_i/b_k)} \right)^{M^*_{k,j}} \right)^{-1} \\ &= \delta_{2,x^*}^{-r'_i} \left(\prod_{j=2,\ldots,n^*} (g^a)^{M^*_{i,j} \cdot y'_j} \right) \left(\prod_{j=1,\ldots,n^*} g^{a^j \cdot s \cdot M^*_{i,j}} \right) \left(\prod_{j=1,\ldots,n^*} g^{a^j \cdot s \cdot M^*_{i,j}} \right)^{-1} \\ &\quad \left(g^{b_i \cdot s}\right)^{-z_{x^*}} \\ &\cdot \left(\prod_{k \in R_i} \prod_{j=1,\ldots,n^*} \left(g^{a^j \cdot s \cdot (b_i/b_k)} \right)^{M^*_{k,j}} \right)^{-1} \\ &= \delta_{2,x^*}^{-r'_i} g^{a \lambda_i} \left(\prod_{j=1,\ldots,n^*} g^{a^j \cdot s \cdot M^*_{i,j}} \right)^{-1} \left(g^{b_i \cdot s}\right)^{-z_{x^*}} \left(\prod_{k \in R_i} \prod_{j=1,\ldots,n^*} \left(g^{a^j \cdot s \cdot (b_i/b_k)} \right)^{M^*_{k,j}} \right)^{-1} \end{aligned}$$

$$= g^{a\lambda_i} g^{z_{x^*}\cdot(-r_i')} g^{b_i\cdot s\cdot(-z_{x^*})} \left(\prod_{i\in X} g^{a\cdot M_{i,1}^*/b_i + a^2\cdot M_{i,2}^*/b_i + \cdots + a^{n^*}\cdot M_{i,n^*}^*/b_i}\right)^{-r_i'}$$

$$\cdot \left(\prod_{j=1,\ldots,n^*} g^{a^j\cdot s\cdot M_{i,j}^*}\right)^{-1} \left(\prod_{k\in R_i}\prod_{j=1,\ldots,n^*} \left(g^{a^j\cdot s\cdot(b_i/b_k)}\right)^{M_{k,j}^*}\right)^{-1}$$

$$= g^{a\lambda_i}\delta_{2,x^*}^{-r_i'-s\cdot b_i} = g^{a\lambda_i}\delta_{2,x^*}^{-r_i} = g^{a\lambda_i} H_3\left(x^*\right)^{-r_i} = g^{a\lambda_i} H_3\left(\rho^*\left(i\right)\right)^{-r_i}, C_i^*$$

$$= g^{r_i'+s\cdot b_i} = g^{r_i}.$$

However, if $T \in_R \mathbb{G}_T$, the challenge ciphertext is independent of the bit b in the view of $\mathcal{A}$.

(5) *Phase 2.* Same as Phase 1.

(6) *Guess.* $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$. If $b' = b$, $\mathcal{C}$ outputs 1 (i.e., deciding $T = e(g,g)^{a^{q+1}\cdot s}$); otherwise, $\mathcal{C}$ outputs 0 (i.e., deciding $T \in_R \mathbb{G}_T$).

This completes the simulations. In what follows, we present the probability analysis. We first analyze the simulations of the random oracles. The simulations of the oracles are perfect except H_1 and H_2. Let H_1^* and H_2^* be the events that $\mathcal{A}$ has queried (m_b, β^*) to H_1 and $R^* = e(g,g)^{\alpha\cdot s}$ to H_2 before the challenge phase, respectively, where $b \in \{0,1\}$, β^* are chosen by $\mathcal{C}$ in the challenge phase. Except for the aforementioned cases, the simulations of H_1 and H_2 are perfect. We denote by $Adv_{H_1^*,\mathcal{A}}^{TCR}$ the probability of $\mathcal{A}$ in querying (m_b, β^*) from H_1 before the challenge phase. Similarly, we have $Adv_{H_2^*,\mathcal{A}}^{TCR}$.

In the simulation of the private key extraction, the responses to $\mathcal{A}$ are perfect. As to the simulation of the re-encryption key extraction, the responses to $\mathcal{A}$ are also perfect except for the case where the re-encryption key is randomly generated. It can be seen that rk_1, rk_2, rk_3 and R_x (which are generated by $\mathcal{C}$) can take the form of the corresponding components of the valid re-encryption key, respectively. Hence, the indistinguishability between the random re-encryption key and the valid one depends on the indistinguishability between the encryption generated by $\mathcal{C}$ and the one constructed in the real scheme. If $\mathcal{A}$ can distinguish the previous encryptions, $\mathcal{C}$ can break the decisional q-parallel BDHE problem by using $\mathcal{A}$.

As for the simulation given in the challenge phase, it is perfect as well.

In the simulation of the re-encryption, the responses to $\mathcal{A}$ are perfect with the exception that $\mathcal{A}$ submits a valid original ciphertext, which is generated without issuing the query to H_1. We denote by $Pr[ReEncErr]$ the probability of the previous exception. Then, we have $Pr[ReEncErr] \leq \frac{q_{re}}{p}$, where q_{re} is the total number of re-encryption queries.

In the simulation of the decryption, it might be possible that $\mathcal{C}$ cannot provide a decryption for a valid ciphertext. Suppose $\mathcal{A}$ can generate a valid ciphertext without querying $e(g,g)^{\alpha\cdot s}$ to H_2, where $s = H_1(m,\beta)$. Let $valid$ be the event that the original ciphertext or the re-encrypted ciphertext is valid, $QueryH_1$ be the event that $\mathcal{A}$ has queried (m,β) to H_1, and $QueryH_2$ be the event that $\mathcal{A}$ has queried $e(g,g)^{\alpha\cdot s}$ to H_2. From the simulation, we have

$$Pr[valid|\neg QueryH_2] \leq Pr[QueryH_1|\neg QueryH_2] + Pr[valid|\neg QueryH_1 \wedge \neg QueryH_2]$$
$$\leq \frac{q_{H_1}}{2^{2k}} + \frac{1}{p},$$

and similarly, we have $Pr[valid|\neg QueryH_1] \leq \frac{q_{H_2}}{2^{2k}} + \frac{1}{p}$, where q_{H_1} and q_{H_2} are the maximum number of random oracle queries to H_1 and H_2. Let $Pr[DecErr]$ be the probability that the event $valid|(\neg QueryH_1 \vee \neg QueryH_2)$ occurs, then we have $Pr[DecErr] \leq (\frac{q_{H_1}+q_{H_2}}{2^{2k}} + \frac{2}{p})\cdot(q_{d1} + q_{d2})$, where q_{d2} and q_{d1} denote the total numbers of original ciphertext decryption queries and re-encrypted ciphertexts decryption queries.

Let Bad denote the event that $(H_1^*|\neg H_2^*) \vee H_2^* \vee ReEncErr \vee DecErr$. Then we have

$$\epsilon_1 = |Pr[b' = b] - \frac{1}{2}| \leq \frac{1}{2}Pr[Bad] = \frac{1}{2}Pr[(H_1^*|\neg H_2^*) \vee H_2^* \vee ReEncErr \vee DecErr]$$
$$\leq \frac{1}{2}(Adv_{H_2^*,\mathcal{A}}^{TCR} + \frac{q_{H_1} + (q_{H_1} + q_{H_2})\cdot(q_{d_1} + q_{d_2})}{2^{2k}} + \frac{2(q_{d_1} + q_{d_2}) + q_{re}}{p}).$$

Therefore, we have

$$
\begin{aligned}
Adv_{\mathcal{A}}^{D-q-parallelBDHE} &\geq \frac{1}{q_{H_2}}(Adv_{H_2^*,A}^{TCR}) \\
&\geq \frac{1}{q_{H_2}}(2\epsilon_1 - \frac{q_{H_1} + (q_{H_1} + q_{H_2}) \cdot (q_{d_1} + q_{d_2})}{2^{2k}} - \frac{2(q_{d_1} + q_{d_2}) + q_{re}}{p}).
\end{aligned}
$$

From the simulation, the running time of $\mathcal{C}$ is bound by

$$
\begin{aligned}
t' \leq t &+ O(1)(q_{H_1} + q_{H_2} + q_{H_3} + q_{H_4} + q_{H_5} + q_{H_6} + q_{sk} + q_{rk} + q_{re} + q_{d2} + q_{d1}) \\
&+ t_e(q_{sk}O(n^{*2}) + (q_{rk} + q_{re})O(f) + (q_{d2} + q_{d1})O(l) + q_{H_1}(q_{re} + q_{d2} + q_{d1})O(1)) \\
&+ t_p((q_{re} + q_{d2} + q_{d1})O(l)),
\end{aligned}
$$

where q_{H_i} denotes the total number of random oracle queries to H_i $(i \in \{1,2,3,4,5,6\})$, q_{sk} and q_{rk} denote the total numbers of private key extraction queries and re-encryption key extraction queries, t_e denotes the running time of an exponentiation in group $\mathbb{G}$, t_p denotes the running time of a pairing in group $\mathbb{G}_T$, t is the running time of $\mathcal{A}$, and l is the number of rows of matrix.

IND-sAS-CCA-Re Security.

Suppose there exists an adversary $\mathcal{A}$ who can break the IND-sAS-CCA-Re security of our scheme. We then construct a reduction algorithm $\mathcal{C}$ to play the decisional q-parallel BDHE problem.

$\mathcal{C}$ takes in $(p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow BSetup(1^k)$ and a q-parallel BDHE instance y and $T \in \mathbb{G}_T$, where T is either equal to $e(g,g)^{a^{q+1} \cdot s}$ or to $T' \in_R \mathbb{G}_T$.

(1) *Initialization.* The same as the previous proof.
(2) *Setup.* The same as the previous proof.
(3) *Phase 1.* The same as the previous proof but with constraints defined in Definition 5.
(4) *Challenge.* $\mathcal{A}$ outputs (M, ρ), m_0 and m_1 to $\mathcal{C}$. $\mathcal{C}$ chooses $b \in_R \{0,1\}$ and responds as follows.

   (a) Run $C_{(M,\rho)} \leftarrow Enc((M,\rho), m_b)$ as in the real scheme, and keep $((M,\rho)$, A_1, A_3, (B_1, C_1), $\ldots$, (B_l, C_l), $D)$. Here the component A_2 is excluded. Note that M is an $l \times n$ matrix.

   (b) Find an attribute set S such that $S \models (M, \rho)$ (note that it is possible that S can be found in SK^{List}), and choose $\beta'^*, \delta^* \in_R \{0,1\}^k$. Issue an H_5 query on δ^* to obtain ξ_2^*. Note that in step (a), the query (m_b, β) must be issued to H_1 such that the tuple (m_b, β, s') is already stored in H_1^{List}, where $\beta \in \{0,1\}^k, s' \in \mathbb{Z}_p^*$. Then recover $(m_b$, $\beta, s')$ from H_1^{List}, and set $A_4^* = (e(g^a, g^{a^q}) \cdot e(g, g^{\alpha'}))^{s' \cdot \xi_2^*}$.

   (c) For each row i of M^* (an $l^* \times n^*$ matrix), set $x^* = \rho^*(i)$, issue an H_3 query on x^* to obtain the tuple $(x^*, z_{x^*}, \delta_{2,x^*})$. Choose $y'_2, \ldots, y'_{n^*}, r'_1, \ldots, r'_{l^*} \in_R \mathbb{Z}_p^*$, for all $i \in \{1, \ldots, l^*\}$, denote R_i as the set of all $i \neq k$ such that $\rho^*(i) = \rho^*(k)$. Set

$$
\begin{aligned}
B_i'^* = \delta_{2,x^*}^{-r'_i} \cdot &\left(\prod_{j=2,\ldots,n^*} g^{a \cdot M^*_{i,j} \cdot y'_j} \right) \cdot g^{b_i \cdot s \cdot (-z_{x^*})}. \\
&\left(\prod_{k \in R_i} \prod_{j=1,\ldots,n^*} \left(g^{a^j \cdot s \cdot (b_i/b_k)} \right)^{M^*_{k,j}} \right)^{-1}, \\
C_i'^* = g^{r'_i + s \cdot b_i}&.
\end{aligned}
$$

   (d) Choose $A_1'^* \in_R \{0,1\}^{2k}$, implicitly define $H_2(T \cdot e(g^s, g^{\alpha'})) = A_1'^* \oplus (\delta^* || \beta'^*)$, and set $A_2'^* = g^s$.

   (e) Issue an H_6 query on $(A_1'^*, A_2'^*, (B_1'^*, C_1'^*), \ldots, (B_{l^*}'^*, C_{l^*}'^*), S, (M^*, \rho^*))$ to obtain the tuple $(A_1'^*, A_2'^*, (B_1'^*, C_1'^*), \ldots, (B_{l^*}'^*, C_{l^*}'^*), S, (M^*, \rho^*), \xi_3^*, \delta_4^*)$, and define $D'^* = (g^s)^{\xi_3^*}$.

   (f) Output the challenge re-encrypted ciphertext $C_{(M^*,\rho^*)}^{R*} = (S, (M, \rho), (M^*, \rho^*), A_1$, $A_3, (B_1, C_1), \ldots, (B_l, C_l), D, A_4^*, A_1'^*, A_2'^*, (B_1'^*, C_1'^*), \ldots, (B_{l^*}'^*, C_{l^*}'^*), D'^*)$ to $\mathcal{A}$.

If $T = e(g,g)^{a^{q+1}\cdot s}$, $C^{R*}_{(M^*,\rho^*)}$ is a valid ciphertext. The components corresponding to $C_{(M,\rho)}$ are valid. Because $C_{(M,\rho)}$ is re-encrypted to $C^{R*}_{(M^*,\rho^*)}$ under a valid re-encryption key $rk_{S\to(M^*,\rho^*)}$ $(S \models (M,\rho))$, the re-encryption must be valid, that is, the construction of A_4^* is valid. With the same analysis technique given in the previous proof, it is not difficult to see that the rest of components are valid as well. If $T \in_R \mathbb{G}_T$, the challenge ciphertext is independent of the bit b in the view of $\mathcal{A}$.

(5) *Phase 2.* The same as Phase 1 .

(6) *Guess.* $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$. If $b' = b$, $\mathcal{C}$ outputs 1 (i.e., deciding $T = e(g,g)^{a^{q+1}\cdot s}$); otherwise, $\mathcal{C}$ outputs 0 (i.e., deciding $T \in_R \mathbb{G}_T$).

The probability analysis is identical to that of the previous proof except that we should take the even H_5^* into account when analyzing the event Bad, where H_5^* denotes the event that $\mathcal{A}$ has queried δ^* to H_5 before the challenge phase. From the simulation, we have $Adv_{\mathcal{A}}^{D-q-parallelBDHE} \geq \frac{1}{q_{H_2}}(Adv^{TCR}_{H_2^*,A}) \geq \frac{1}{q_{H_2}}(2\epsilon_2 - \frac{q_{H_1}+(q_{H_1}+q_{H_2})\cdot(q_{d_1}+q_{d_2})}{2^{2k}} - \frac{q_{H_5}}{2^k} - \frac{2(q_{d_1}+q_{d_2})+q_{re}}{p})$. The running time of $\mathcal{C}$ is identical to that of the previous proof.

This completes the proof of Theorem 2. □

5. CONCLUSIONS

In this paper, we proposed a new single-hop unidirectional CP-ABPRE scheme, which supports attribute-based re-encryption with any monotonic access structure, to tackle the open problem left by the existing CP-ABPRE schemes. We also showed that our scheme can be proved IND-sAS-CCA secure in the random oracle model under the decisional q-parallel BDHE assumption.

Removing the ROM. The technique introduced in [37] might be a possible approach to remove random oracles.

We leave this as our future work.

This paper also motivates some interesting open problems, for example, how to construct a CCA-secure CP-ABPRE scheme in the adaptive access structure model, that is, achieving IND-aAS-CCA security.

ACKNOWLEDGEMENTS

This work was supported by the Fundamental Research Funds for the Central Universities, National Natural Science Foundation of China, National Natural Science Foundation of Jiangsu, National Science Foundation for Post-doctoral Scientists of China, National Science Foundation for Post-doctoral Scientists of Jiangsu, Australian Research Council Linkage Project, and Research Grants Council of HKSAR, grant numbers NZ2013306, 61272083, 61300236, BK20130809, 2013M530254, 1302137C, LP120200052, CityU 123511.

REFERENCES

1. Sahai A, Waters B. Fuzzy identity-based encryption. In *Eurocrypt*, vol. 3494, Cramer R (ed.)., Lecture Notes in Computer Science. Springer: Springer-Verlag Berlin, Heidelberg, 2005; 457–473.
2. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS), 2006,* Alexandria, VA, USA. ACM Press: ACM New York, NY, USA, 2006; 89–98.
3. Mambo M, Okamoto E. Proxy cryptosystems: delegation of the power to decrypt ciphertexts. *IEICE Transactions* 1997; **E80-A**(1):54–63.
4. Blaze M, Bleumer G, Strauss M. Divertible protocols and atomic proxy cryptography. In *Proceedings of Advances in Cryptology – EUROCRYPT 1998, International Conference on the Theory and Application of Cryptographic Techniques (Lecture Notes in Computer Science),* Berlin, Vol. 1403, Nyberg K (ed.). Springer: Springer Berlin Heidelberg, 1998; 127–144.
5. Ateniese G, Fu K, Green M, Hohenberger S. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security* 2006; **9**(1):1–30.
6. Biswas P, Towsley D, Ramakrishnan KK, Krishna CM. Performance benefits of non-volatile caches in distributed file systems. *Concurrency: Practice and Experience* 1994; **6**(4):289–323.

DOI: 10.1002/cpe

7. Wang G, Liu Q, Wu J. Achieving fine-grained access control for secure data sharing on cloud servers. *Concurrency and Computation: Practice and Experience* 2011; **23**(12):1443–1464.
8. Weng J, Deng RH, Ding X, Chu C-K, Lai J. Conditional proxy re-encryption secure against chosen-ciphertext attack. In *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security (ASIACCS), March 2009*. ACM Press: Sydney, Australia, 2009; 322–332.
9. Green M, Ateniese G. *Identity-based proxy re-encryption*, In ACNS '07, vol. 4512. Springer, 2007; 288–306.
10. Liang X, Cao Z, Lin H, Shao J. Attribute based proxy re-encryption with delegating capabilities. In *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security (ASIACCS), March 2009,* Sydney, Australia. ACM Press: ACM New York, NY, USA, 2009; 276–286.
11. Chen X, Li J, Susilo W. Efficient fair conditional payments for outsourcing computations. *IEEE Transactions on Information Forensics and Security* 2012; **7**(6):1687–1694.
12. Hanaoka G, Kawai Y, Kunihiro N, Matsuda T, Weng J, Zhang R, Zhao Y. Generic construction of chosen ciphertext secure proxy re-encryption. In *Proceedings of Topics in Cryptology – CT-RSA 2012 – The Cryptographers'Track at the RSA Conference 2012 (Lecture Notes in Computer Science),* Berlin, Vol. 7178, Dunkelman O (ed.). Springer: Springer Berlin Heidelberg, 2012; 349–364.
13. Canetti R, Halevi S, Katz J. Chosen-ciphertext security from identity-based encryption. In *Proceedings of Advances in Cryptology – EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques (Lecture Notes in Computer Science),* Berlin, Vol. 3027, Cachin C, Camenisch J (eds). Springer: Springer Berlin Heidelberg, 2004; 207–222.
14. Canetti R, Krawczyk H, Nielsen JB. Relaxing chosen-ciphertext security. In *Proceedings of Advances in Cryptology – CRYPTO 2003, 23rd Annual International Cryptology Conference (Lecture Notes in Computer Science),* Berlin, Vol. 2729, Boneh D (ed.). Springer: Springer Berlin Heidelberg, 2003; 565–582.
15. Libert B, Vergnaud D. Unidirectional chosen-ciphertext secure proxy re-encryption. In *Proceedings of Public Key Cryptography – PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography (Lecture Notes in Computer Science),* Berlin, Vol. 4939, Cramer R (ed.). Springer: Springer Berlin Heidelberg, 2008; 360–379.
16. Chu CK, Tzeng WG. Identity-based proxy re-encryption without random oracles. In *Proceedings of Information Security, 10th International Conference (Lecture Notes in Computer Science),* Berlin, Vol. 4779, Garay JA, Lenstra AK, Mambo M, Peralta R (eds). Springer: Springer Berlin Heidelberg, 2007; 189–202.
17. Emura K, Miyaji A, Omote K. A timed-release proxy re-encryption scheme. *IEICE Transactions* 2011; **94-A**(8): 1682–1695.
18. Luo S, Hu J, Chen Z. Ciphertext policy attribute-based proxy re-encryption. In *Proceedings of Information and Communications Security – 12th International Conference (Lecture Notes in Computer Science),* Berlin, Vol. 6476, Soriano M, Qing S, López J (eds). Springer: Springer Berlin Heidelberg, 2010; 401–415.
19. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P), May 2007*. IEEE Computer Society Press: Oakland, California, USA, 2007; 321–334.
20. Cheung L, Newport CC. Provably secure ciphertext policy ABE. In *Proceedings of the 2007 ACM Conference on Computer and Communications Security (CCS), October 2007,* Alexandria, Virginia, USA. ACM Press: ACM New York, NY, USA, 2007; 456–465.
21. Goyal V, Jain A, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption. In *Proceedings of Automata, Languages and Programming, 35th International Colloquium, ICALP: Part II – Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations (Lecture Notes in Computer Science),* Berlin, Vol. 5126, Aceto L, Damgård I, Goldberg LA, Halldórsson MM, Ingólfsdóttir A, Walukiewicz I (eds). Springer: Springer-Verlag Berlin, Heidelberg, 2008; 579–591.
22. Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. In *Proceedings of Public Key Cryptography – PKC 2011 – 14th International Conference on Practice and Theory in Public Key Cryptography (Lecture Notes in Computer Science),* Berlin, Vol. 6571, Catalano D, Fazio N, Gennaro R, Nicolosi A (eds). Springer: Springer Berlin Heidelberg, 2011; 53–70.
23. Lewko AB, Okamoto T, Sahai A, Takashima K, Waters B. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. In *Proceedings of Advances in Cryptology – EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Lecture Notes in Computer Science),* Berlin, Vol. 6110, Gilbert H (ed.). Springer: Springer Berlin Heidelberg, 2010; 62–91.
24. Lewko AB, Waters B. New proof methods for attribute-based encryption: achieving full security through selective techniques. In *Proceedings of Advances in Cryptology – CRYPTO 2012 – 32nd Annual Cryptology Conference (Lecture Notes in Computer Science),* Berlin, Vol. 7417, Safavi-Naini R, Canetti R (eds). Springer: Springer Berlin Heidelberg, 2012; 180–198.
25. Attrapadung N, Herranz J, Laguillaumie F, Libert B, Panafieu ED, Rafols C. Attribute-based encryption schemes with constant-size ciphertexts. *Theoretical Computer Science* 2012; **422**(0):15–38.
26. Waters B. Functional encryption for regular languages. In *Proceedings of Advances in Cryptology – CRYPTO 2012 – 32nd Annual Cryptology Conference (Lecture Notes in Computer Science),* Berlin, Vol. 7417, Safavi-Naini R, Canetti R (eds). Springer: Springer Berlin Heidelberg, 2012; 218–235.
27. Li J, Huang Q, Chen X, Chow SSM, Wong DS, Xie D. Multi-authority ciphertext-policy attribute-based encryption with accountability. In *Proceedings of the 2011 ACM Symposium on Information, Computer and Communications*

DOI: 10.1002/cpe

Security (ASIACCS), March 2011, Hong Kong, China, Cheung BSN, Hui LCK, Sandhu RS, Wong DS (eds). ACM Press: ACM New York, NY, USA, 2011; 386–390.

28. Ivan AA, Dodis Y. Proxy Cryptography Revisited. In *Proceedings of the Network and Distributed System Security Symposium (NDSS), 2003*. The Internet Society Press: San Diego, California, USA, 2003.
29. Liang K, Qiong H, Schlegel R, Wong DS, Tang C. A conditional proxy broadcast re-encryption scheme supporting timed-release. In *Proceedings of Information Security Practice and Experience – 9th International Conference*, Vol. 7863, Deng RH, Feng T (eds). Springer. Lecture Notes in Computer Science, 2013; 132–146.
30. Canetti R, Hohenberger S. Chosen-ciphertext secure proxy re-encryption. In *Proceedings of the 2007 ACM Conference on Computer and Communications Security (CCS), October 2007,* Alexandria, Virginia, USA, Ning P, di Vimercati SDC, Syverson PF (eds). ACM Press: ACM New York, NY, USA, 2007; 185–194.
31. Isshiki T, Nguyen MH, Tanaka K. Proxy re-encryption in a stronger security model extended from CT-RSA 2012. In *Proceedings of Topics in Cryptology – CT-RSA 2013 – The Cryptographers' Track at the RSA Conference 2013 (Lecture Notes in Computer Science)*, Vol. 7779, Dawson (ed.). Springer: Berlin, 2013; 277–292.
32. Mizuno T, Doi H. Hybrid proxy re-encryption scheme for attribute-based encryption. In *Inscrypt*, vol. 6151, Bao F, Yung M, Lin D, Jing J (eds)., Lecture Notes in Computer Science. Springer: Springer Berlin Heidelberg, 2009; 288–302.
33. Liang K, Fang L, Susilo W, Wong DS. A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security, 2013; 552–559. IEEE.
34. Beimel A. Secure schemes for secret sharing and key distribution. *Ph.D. Thesis*, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
35. Cramer R, Shoup V. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. *SIAM J. Comput.* 2004; **33**(1):167–226.
36. Fujisaki E, Okamoto T. Secure integration of asymmetric and symmetric encryption schemes. *J. Cryptology* 2013; **26**(1):80–101.
37. Liang K, Liu Z, Tan X, Wong DS, Tang C. A CCA-secure identity-based conditional proxy re-encryption without random oracles. In *Icisc*, vol. 7839, Kwon T, Lee MK, Kwon D (eds)., Lecture Notes in Computer Science. Springer: Springer Berlin Heidelberg, 2012; 231–246.

DOI: 10.1002/cpe