Future Generation Computer Systems 52 (2015) 95–108

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A secure and efficient Ciphertext-Policy Attribute-Based Proxy Re-Encryption for cloud data sharing☆

Kaitai Liang [a,*], Man Ho Au [b], Joseph K. Liu [c,*], Willy Susilo [d], Duncan S. Wong [a], Guomin Yang [d], Yong Yu [d], Anjia Yang [a]

[a] Department of Computer Science, City University of Hong Kong, Hong Kong Special Administrative Region

[b] Department of Computing, Hong Kong Polytechnic University, Hong Kong Special Administrative Region

[c] Institute for Infocomm Research, A*STAR, Singapore

[d] Centre for Computer and Information Security Research, School of Computer Science and Software Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

HIGHLIGHTS

- This paper proposes a new Ciphertext-Policy Attribute-Based Proxy Re-Encryption scheme.
- The scheme is proved adaptively chosen ciphertext secure by leveraging dual system encryption technology and selective proof technique.
- The paper also proposes an improvement for re-encryption key generation and re-encryption phases so as to reduce computational and communication cost.

ARTICLE INFO

Article history:
Received 30 June 2014
Received in revised form
11 November 2014
Accepted 23 November 2014
Available online 3 December 2014

Keywords:
Ciphertext-policy attribute-based encryption
Ciphertext-policy attribute-based proxy re-encryption
Adaptive chosen-ciphertext security

ABSTRACT

Proxy Re-Encryption (PRE) is a useful cryptographic primitive that allows a data owner to delegate the access rights of the encrypted data stored on a cloud storage system to others without leaking the information of the data to the honest-but-curious cloud server. It provides effectiveness for data sharing as the data owner even using limited resource devices (e.g. mobile devices) can offload most of the computational operations to the cloud. Since its introduction many variants of PRE have been proposed. A Ciphertext-Policy Attribute-Based Proxy Re-Encryption (CP-ABPRE), which is regarded as a general notion for PRE, employs the PRE technology in the attribute-based encryption cryptographic setting such that the proxy is allowed to convert an encryption under an access policy to another encryption under a new access policy. CP-ABPRE is applicable to many network applications, such as network data sharing. The existing CP-ABPRE systems, however, leave how to achieve adaptive CCA security as an interesting open problem. This paper, for the first time, proposes a new CP-ABPRE to tackle the problem by integrating the dual system encryption technology with selective proof technique. Although the new scheme supporting any monotonic access structures is built in the composite order bilinear group, it is proven adaptively CCA secure in the standard model without jeopardizing the expressiveness of access policy. We further make an improvement for the scheme to achieve more efficiency in the re-encryption key generation and re-encryption phases.

1. Introduction

Attribute-Based Encryption (ABE) [1,2], which is a generalization of Public Key Encryption (PKE), provides flexibility of data sharing for system users in the sense that a data encryptor is allowed to specify some descriptive values x for an encryption and thus, the encryption can be decrypted successfully by a secret key associated with some descriptive values y matching x. ABE has many network applications, such as cloud storage systems, and audit log applications [1]. It often has two different classifications:

☆ This is the full version of the paper Liang et al. (2014) published in the 10th International Conference on Information Security Practice and Experience (ISPEC 2014).

* Corresponding authors.
E-mail addresses: kliang4-c@my.cityu.edu.hk (K. Liang), csallen@comp.polyu.edu.hk (M.H. Au), ksliu@i2r.a-star.edu.sg (J.K. Liu), wsusilo@uow.edu.au (W. Susilo), duncan@cityu.edu.hk (D.S. Wong), gyang@uow.edu.au (G. Yang), yyong@uow.edu.au (Y. Yu), ayang3-c@my.cityu.edu.hk (A. Yang).

http://dx.doi.org/10.1016/j.future.2014.11.016
0167-739X/

one is Key-Policy ABE (KP-ABE), and the other is Ciphertext-Policy ABE (CP-ABE). In a KP-ABE system, ciphertexts are associated with attribute sets and secret keys are associated with access policies. However, in a CP-ABE scheme ciphertexts are related to access policies, and attribute sets are tagged with secret keys. Note this paper mainly deals with the case of CP-ABE.

In some social network settings, a system user, say Alice, might choose to share her profile under a specified policy such that other users satisfying the policy can gain access to the profile. Alice might encrypt her profile under an access policy AP_1 before storing to the social network server. The system users satisfying AP_1 then can decrypt the profile from the cloud, and next access the data by using the corresponding secret keys. This sharing pattern, nonetheless, does not scale well when the access policy needs to be updated frequently. Suppose the policy above needs to be updated to a new policy AP_2, Alice then should generate a new encryption for AP_2 accordingly. If Alice does not back up the profile (in the plain format) locally, then she has to download the ciphertext and recovers the profile first. If the access policy is renewed N times, Alice is required to construct N new encryptions. This might not be desirable in practice as the workload of Alice is increased linearly in N. Besides, if Alice is either off-line or using some resource-limited devices, which cannot afford heavy computational cost incurred by decryption and encryption, the sharing might not be fulfilled effectively.

To make data sharing more effectively, we may leverage the technology of Proxy Re-Encryption (PRE). The notion of PRE is introduced by Mambo and Okamoto [3], and further formally defined by Blaze, Bleumer and Strauss [4]. PRE is an interesting extension of Public Key Encryption (PKE) providing the delegation of decryption rights. Specifically, it allows a *semi-trusted* proxy to transform a ciphertext intended for Alice into another ciphertext of the same plaintext intended for another system user, say Bob, without revealing knowledge of the secret keys and the underlying plaintext. It is a useful cryptographic primitive for many applications, such as secure distributed files systems [5] and email forwarding [4].

To integrate the PRE technology in the ABE cryptographic setting, Liang et al. [6] first defined the notion of Ciphertext-Policy Attribute-Based PRE (CP-ABPRE), and proposed a concrete CP-ABPRE system enabling the proxy to transform an encryption under a specified access policy into another encryption under a new access policy. We refer to this special functionality as *attribute-based re-encryption* in this paper. By using the technology of CP-ABPRE, Alice can share the data more effectively. She first generates a re-encryption key from her own attribute set to a new access policy AP_2, and next uploads the key to the cloud such that the cloud server then can convert the original encryption under AP_1 to a new encryption under AP_2. The server, nevertheless, cannot learn the data during the conversion of ciphertexts.

Although CP-ABPRE systems explore the applications of PRE, they leave us some interesting open problems. We here show that it is very important to solve the problems. As far as we know, all the existing CP-ABPRE schemes in the literature are secure against *selective chosen-plaintext attacks* (selective CPA) only except [7] which is *selective chosen-ciphertext attacks* (selective CCA) secure. We state that CPA security might not be sufficient enough in an open network setting as it only guarantees the secrecy of data which only holds against "static" adversaries. Nevertheless, in a real network scenario, there might exist "active" adversaries trying to tamper an encryption in transit and next observing its decryption so as to obtain useful information related to the underlying data. Accordingly, a CP-ABPRE system being secure against CCA is needed as CCA security not only helps the system preclude the above subtle attacks but also enables the system to be further developed and next securely "embedded" to a large protocol/system implementing in arbitrary network environments. In addition, a CP-ABPRE system with selective security, which limits an adversary to choose an attack target before playing a security game, might not scale well in practice as well. This is so because a realistic adversary is able to adaptively choose his attack target when attacking a cryptosystem. Therefore, an *adaptively CCA secure* CP-ABPRE scheme is extremely desirable in most practical network applications.

The expressiveness of access policy is another crucial factor for a practical CP-ABPRE system. In practice an access policy should be embedded with *AND*, *OR* gates, and even more meaningful expression. Nevertheless, all the existing CP-ABPRE schemes only support access policy with *AND* gates operating over attributes. Although [7] supports more expressive policy by leveraging LSSS, it is built in the random oracle model. This might not be practical as the system being secure in the random oracle model might suffer from potential security attacks in practical setting. Thus it is desirable to propose a CP-ABPRE system supporting expressive access policy without random oracles.

1.1. Our contributions

This work first formalizes the notion of adaptive CCA security for CP-ABPRE systems. When compared with the selective CPA security notion, our new notion enables an adversary to commit to a target access policy in the challenge phase, and to gain access to re-encryption and decryption oracles additionally. To tackle the open problems left by the existing CP-ABPRE schemes, this paper further proposes a novel single-hop unidirectional CP-ABPRE system by integrating the dual system encryption technology with the selective proof technique. In addition, the new system supports any monotonic access policy such that system users are allowed to fulfill more flexible delegation of decryption rights. Despite our scheme is built in the composite order bilinear group, it is proven adaptively CCA secure in the standard model.

Differences between this version and the conference one. Compared with the conference version, the following contents are added to this version. First of all, we make an efficiency improvement for our conference version. After being improved, the new scheme enjoys more efficiency in terms of communication and computation with respect to the re-encryption key generation and re-encryption phases. Specifically, the improvement saves $O(a)|\mathbb{G}|$ and $O(a)c_e$ in terms of communication and computation cost in the re-encryption key generation phase; as to the re-encryption phase, it reduces the number of pairings (computational cost) from $O(a)$ to constant. Furthermore, we present the formal security analysis for our basic construction (presented in conference version) in the composite order bilinear group by integrating the dual system encryption technology with selective proof technique. We also describe the technical construction roadmap and difficulties before presenting the basic construction, make correctness analysis, and give a discussion on the properties acquired by the scheme. We further explore some practical applications for our system, such as secure cloud storage data sharing, introduce preliminaries for building blocks, data structure, complexity assumptions used throughout the paper. We finally update the related work as well.

1.2. Related work

Below we mainly review some classic ABE systems related to our current work. Following the introduction of ABE due to Sahai and Waters [2], Goyal et al. [1] proposed the first KP-ABE system, in which ciphertexts are associated with attributes, and secret keys are associated with access policies (over attributes). Later, Bethencourt, Sahai and Waters [8] defined a complementary notion, i.e. CP-ABE, in which ciphertexts are associated with access policies and secret keys are related to attributes. Cheung

Table 1
Comparison with [6,29,30].

Sch.	Public/secret key size	Ciph. size	Re-Enc cost	Adaptive security	CCA security
[6]	$\mathcal{O}(u)/\mathcal{O}(u)$	$\mathcal{O}(u)$	$\mathcal{O}(u)\cdot c_p$	✗	✗
[29]	$\mathcal{O}(u^2)/\mathcal{O}(u)$	$\mathcal{O}(u)$	$\mathcal{O}(u)\cdot c_p$	✗	✗
[30]	$\mathcal{O}(u)/\mathcal{O}(u)$	$\mathcal{O}(u)$	$\mathcal{O}(1)\cdot c_e+\mathcal{O}(u)\cdot c_p$	✗	✗
Ours	$\mathcal{O}(u)/\mathcal{O}(a)$	$\mathcal{O}(l)$	$\mathcal{O}(a)\cdot c_e+\mathcal{O}(a)\cdot c_p$	✓	✓

and Newport [9] proposed a provably secure CP-ABE scheme which supports *AND* gates over attributes. Goyal et al. [10] constructed a CP-ABE scheme but with large key size. Waters [11] designed efficient and expressive CP-ABE systems supporting any monotonic access structure. Attrapadung et al. [12] proposed an efficient CP-ABE for threshold access policy with constant-size ciphertexts. Later on, Waters [13] proposed the first deterministic finite automata-based functional encryption system in which access policy can be expressed by arbitrary-size regular language. Note that there are also some variants of traditional ABE in the literature, such as [14–16].

The aforementioned schemes, nonetheless, are only selective secure (except for [8] being proven in the generic group model). To convert one of the CP-ABE systems proposed in [11] to achieve fully security, Lewko et al. [17] adapted the dual system encryption technology to the ABE cryptographic setting. But their conversion yields some loss of expressiveness (which they called "one-use restriction"). Later, Lewko and Waters [18] introduced a new method to capture full security without jeopardizing the expressiveness by employing the selective proof technique into the dual system encryption technology. Inspired by [18,11], this paper focuses on constructing a CP-ABPRE with adaptive CCA security in the standard model.

The concept of decryption rights delegation is first introduced in [3]. Later on, Blaze, Bleumer and Strauss [4] formally defined the notion of PRE. PRE can be classified as: unidirectional and bidirectional PRE, and single-hop and multi-hop PRE, where the corresponding definitions are given in [5]. This present work deals with the single-hop unidirectional case. Since its introduction many variants of PRE systems have been proposed, such as [5,19–28].

To employ PRE technology in the ABE cryptographic setting, Liang et al. [6] first defined CP-ABPRE, and further extended [9] to support proxy re-encryption. This seminal CP-ABPRE system provides *AND* gates over positive and negative attributes. Later on, Luo et al. [29] proposed an extension of [6], 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 [30] 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 selectively CPA secure, and their access policies are lack of expressiveness due to supporting *AND* gates over attributes only. Thus an adaptively CCA-secure CP-ABPRE scheme with more expressive access policy remains open. In TCC 2012, Chandran, Chase and Vaikuntanathan [31] proposed an obfuscation for functional re-encryption with collusion resistant property. Recently, a CCA-secure CP-ABPRE was proposed in [7], in which the scheme is proven in the random oracle model. In [32], a CP-ABPRE system was built and proven secure against CCA in the standard model.

We compare our work with all existing CP-ABPRE schemes in terms of public/secret key size, ciphertext size, re-encryption cost and security (shown in Table 1). We let l be the number of row of a matrix in an access policy, a be the number of attributes embedded in a user's secret key and u be the total number of attributes used in the system. In the worst case, an access policy and a user's secret key might be embedded with all system attributes, that is $l = a = u$. Thus we have $l, a \leq u$. We use c_e and c_p to denote the computational cost of an exponentiation and a bilinear pairing.

2. Definitions and security models

We review some data structures used in our construction and the definition of CP-ABPRE systems, and next define the adaptive CCA security notion.

Definition 1 (*Access Structures [33]*)**.** 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 $\mathcal{P}$, i.e., $\mathbb{AS} \subseteq 2^{\mathcal{P}} \setminus \emptyset$. The sets in $\mathbb{AS}$ are called the authorized sets, and the sets not in $\mathbb{AS}$ are called the unauthorized sets.

In 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 [33], any monotone access structure can be represented by a linear secret sharing scheme.

Definition 2 (*Linear Secret Sharing Schemes (LSSS) [11]*)**.** 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 A with l rows and n columns called the share-generating matrix for Π. For all $j = 1, \ldots, l$, the jth row of A is labeled by a party $\rho(j)$, 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 $A \cdot v$ is the vector of l shares of the secret s according to Π. The share $(A \cdot v)_j$ belongs to party $\rho(j)$. For any unauthorized set, no such constants exist. We use LSSS matrix (A, ρ) to represent an access policy in this paper.

Note that every LSSS according to the above definition achieves the *linear reconstruction* property [33]. Suppose that 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 $S \models (A, \rho)$) be any authorized set, and let $I \subseteq \{1, 2, \ldots, l\}$ be defined as $I = \{j : \rho(j) \in S\}$. There will exist constants $\{w_j \in \mathbb{Z}_p\}_{j \in I}$ such that $\sum_{j \in I} w_j \cdot \lambda_j = s$ if $\{\lambda_j\}$ are valid shares of any secret s according to Π. Note that as shown in [33] $\{w_j\}$ can be found (with knowledge of A and I) in time polynomial in the size of the share-generating matrix A. In this paper we employ LSSS matrix over $\mathbb{Z}_N$ ($N = p_1 p_2 p_3$), and assume that for an unauthorized set the corresponding rows of the matrix do not include the vector $(1, 0, \ldots, 0)$ in their span modulo p_2.

2.1. Definition of CP-ABPRE

We review the definition of single-hop unidirectional CP-ABPRE [6,29].

Definition 3. A Single-Hop Unidirectional Ciphertext-Policy Attribute-Based Proxy Re-Encryption (CP-ABPRE) scheme consists of the following 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 describing the key, output a secret key sk_S for S.
3. $rk_{S\rightarrow(A',\rho')} \leftarrow ReKeyGen(param, sk_S, (A', \rho'))$: on input $param$, sk_S, and an access structure (A', ρ') for attributes over $\mathcal{U}$, output a re-encryption key $rk_{S\rightarrow(A',\rho')}$ which can be used to transform a ciphertext under (A, ρ) to another ciphertext under (A', ρ'), where $S \models (A, \rho)$, $S \not\models (A', \rho')$, (A, ρ) and (A', ρ') are two disjoint access structures. Note by two disjoint access structures we mean for any attribute x satisfies (A, ρ), x does not satisfy (A', ρ').
4. $C \leftarrow Encrypt(param, (A, \rho), m)$: on input $param$, (A, ρ), and a message $m \in \{0, 1\}^k$, output an original ciphertext C which can be further re-encrypted. Note (A, ρ) is implicitly included in the ciphertext.
5. $C_R \leftarrow ReEnc(param, rk_{S\rightarrow(A',\rho')}, C)$: on input $param$, $rk_{S\rightarrow(A',\rho')}$, and an original ciphertext C under (A, ρ), output a re-encrypted ciphertext C_R under (A', ρ') if $S \models (A, \rho)$ or a symbol $\perp$ indicating either C is invalid or $S \not\models (A, \rho)$. Note C_R cannot be further re-encrypted.
6. $m \leftarrow Dec(param, sk_S, C)$: on input $param$, sk_S, and an original ciphertext C under (A, ρ), output a message m if $S \models (A, \rho)$ or a symbol $\perp$ indicating either C is invalid or $S \not\models (A, \rho)$.
7. $m \leftarrow Dec_R(param, sk_S, C_R)$: on input $param$, sk_S, and a re-encrypted ciphertext C_R under (A, ρ), output a message m if $S \models (A, \rho)$ or a symbol $\perp$ indicating either C_R is invalid or $S \not\models (M, \rho)$.

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

$$Dec(param, sk_S, Encrypt(param, (A, \rho), m)) = m;$$

$$Dec_R(param, sk_{S'}, ReEnc(param, ReKeyGen(param, sk_S, (A', \rho')), Encrypt(param, (A, \rho), m))) = m,$$

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

2.2. Security models

All the existing security notions for CP-ABPRE systems are only defined in the selective CPA security model. Below we define the adaptive CCA security notion, which is denoted as IND-CCA.

Definition 4. A single-hop unidirectional CP-ABPRE scheme is IND-CCA secure at original ciphertext if no Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$ can win the game below with non-negligible advantage. In the game, $\mathcal{C}$ is the challenger, k and $\mathcal{U}$ are the security parameter and attribute universe.

1. **Setup**. $\mathcal{C}$ runs $Setup(1^k, \mathcal{U})$ and sends $param$ to $\mathcal{A}$.
2. **Phase 1**.
   (a) Secret key extraction oracle $\mathcal{O}_{sk}(S)$: on input an attribute set S, $\mathcal{C}$ runs $sk_S \leftarrow KeyGen(param, msk, S)$ and returns sk_S to $\mathcal{A}$.
   (b) Re-encryption key extraction oracle $\mathcal{O}_{rk}(S, (A', \rho'))$: on input S, and an access structure (A', ρ'), $\mathcal{C}$ returns $rk_{S\rightarrow(A',\rho')} \leftarrow ReKeyGen(param, sk_S, (A', \rho'))$ to $\mathcal{A}$, where $sk_S \leftarrow KeyGen(param, msk, S)$.
   (c) Re-encryption oracle $\mathcal{O}_{re}(S, (A', \rho'), C)$: on input S, (A', ρ'), and an original ciphertext C under (A, ρ), $\mathcal{C}$ returns $C_R \leftarrow ReEnc(param, rk_{S\rightarrow(A',\rho')}, C)$ to $\mathcal{A}$, where $rk_{S\rightarrow(A',\rho')} \leftarrow ReKeyGen(param, sk_S, (A', \rho'))$, $sk_S \leftarrow KeyGen(param, msk, S)$ and $S \models (A, \rho)$.
   (d) Original ciphertext decryption oracle $\mathcal{O}_{dec}(S, C)$: on input S and an original ciphertext C under (A, ρ), $\mathcal{C}$ returns $m \leftarrow Dec(param, sk_S, C)$ to $\mathcal{A}$, where $sk_S \leftarrow KeyGen(param, msk, S)$ and $S \models (A, \rho)$.
   (e) Re-encrypted ciphertext decryption oracle $\mathcal{O}_{dec_R}(S, C_R)$: on input S and a re-encrypted ciphertext C_R under (A, ρ), $\mathcal{C}$ returns $m \leftarrow Dec_R(param, sk_S, C_R)$, where $sk_S \leftarrow KeyGen(param, msk, S)$ and $S \models (A, \rho)$.

   If the ciphertexts issued to $\mathcal{O}_{re}$, $\mathcal{O}_{dec}$ and $\mathcal{O}_{dec_R}$ are invalid, $\mathcal{C}$ simply outputs $\perp$.
3. **Challenge**. $\mathcal{A}$ outputs two equal length messages m_0 and m_1, and a challenge access structure (A^*, ρ^*) to $\mathcal{C}$. If the following queries

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

   are never made, $\mathcal{C}$ returns $C^* = Encrypt(param, (A^*, \rho^*), m_b)$ to $\mathcal{A}$, where $b \in_R \{0, 1\}$.
4. **Phase 2**. $\mathcal{A}$ continues making queries as in Phase 1 except the following:
   (a) $\mathcal{O}_{sk}(S)$ for any $S \models (A^*, \rho^*)$;
   (b) $\mathcal{O}_{rk}(S, (A', \rho'))$ for any $S \models (A^*, \rho^*)$, and $\mathcal{O}_{sk}(S')$ for any $S' \models (A', \rho')$;
   (c) $\mathcal{O}_{re}(S, (A', \rho'), C^*)$ for any $S \models (A^*, \rho^*)$, and $\mathcal{O}_{sk}(S')$ for any $S' \models (A', \rho')$;
   (d) $\mathcal{O}_{dec}(S, C^*)$ for any $S \models (A^*, \rho^*)$; and
   (e) $\mathcal{O}_{dec_R}(S, C_R)$ for any C_R under (A, ρ), $S \models (A, \rho)$, where C_R is a derivative of C^*. As of [19], the derivative of C^* is defined as follows.
      i. C^* is a derivative of itself.
      ii. If $\mathcal{A}$ has issued a re-encryption key query on $(S^*, (A', \rho'))$ to get $rk_{S^*\rightarrow(A',\rho')}$, and obtained $C_R \leftarrow ReEnc(param, rk_{S^*\rightarrow(A',\rho')}, C^*)$ such that $Dec_R(param, sk_{S'}, C_R) \in \{m_0, m_1\}$, then C_R is a derivative of C^*, where $S^* \models (A^*, \rho^*)$ and $S' \models (A', \rho')$.
      iii. If $\mathcal{A}$ has issued a re-encryption query on $(S, (A', \rho'), C^*)$ and obtained the re-encrypted ciphertext C_R, then C_R is a derivative of C^*, where $S \models (A^*, \rho^*)$.
5. **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 $Adv^{IND\text{-}CCA\text{-}Or}_{CP\text{-}ABPRE,\mathcal{A}}(1^k, \mathcal{U}) = |Pr[b' = b] - \frac{1}{2}|$.

The adaptive CCA security for re-encrypted ciphertext (IND-CCA-Re) is defined as follows.

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

$$Adv^{IND\text{-}CCA\text{-}Re}_{CP\text{-}ABPRE,\mathcal{A}}(1^k, \mathcal{U}) = \Big| Pr[b' = b : (param, msk) \leftarrow Setup(1^k, \mathcal{U}); (m_0, m_1, (A^*, \rho^*), (A, \rho), State) \leftarrow \mathcal{A}^{\mathcal{O}}(param); b \in_R \{0, 1\}; C^*_R \leftarrow ReEnc(param, rk_{S\rightarrow(A^*,\rho^*)}, C); b' \leftarrow A^{\mathcal{O}}(C^*_R, State)] - \frac{1}{2} \Big|,$$

where $State$ is the state information, (A, ρ) and (A^*, ρ^*) are disjoint, (A^*, ρ^*) is the challenge access structure, $S \models (A, \rho)$, $rk_{S\rightarrow(A^*,\rho^*)} \leftarrow ReKeyGen(param, sk_S, (A^*, \rho^*))$, $C \leftarrow Encrypt(param, (A, \rho), m_b)$, $\mathcal{O}_{sk}$, $\mathcal{O}_{rk}$, $\mathcal{O}_{dec}$, $\mathcal{O}_{dec_R}$ are the oracles defined in Definition 4. However, these oracles are restricted by the following constraints. For $\mathcal{O}_{sk}$, any query $S \models (A^*, \rho^*)$ is rejected. There is no restriction to $\mathcal{O}_{rk}$ and $\mathcal{O}_{dec}$ (note invalid ciphertexts issued to $\mathcal{O}_{dec}$ are rejected). If $\mathcal{A}$ queries to $\mathcal{O}_{dec_R}$ on either (S, C^*_R) in which $S \models (A^*, \rho^*)$ or any invalid re-encrypted ciphertext, the oracle outputs $\perp$.

Remarks. Definition 5 implies collusion resistance. This is because if $\mathcal{A}$ can compromise a key sk_{S^*} from either $rk_{S^*\to(A,\rho)}$ or $rk_{S\to(A^*,\rho^*)}$, $\mathcal{A}$ wins the above game with non-negligible probability, where $S \models (A, \rho)$, $S^* \models (A^*, \rho^*)$ and $\mathcal{A}$ obtains sk_S.

3. Preliminaries

We first give a brief introduction for composite order bilinear groups and some complexity assumptions used in our security proof. We next review the target collision resistant hash function, one-time signature system and one-time symmetric encryption system.

Composite order bilinear groups [34]. Let $\mathbb{G}$ and $\mathbb{G}_T$ be the two multiplicative cyclic groups of order $N = p_1p_2p_3$, where p_1, p_2, p_3 are distinct primes. We say that $\mathbb{G}_T$ has an admissible bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ if the following properties hold: (1) *Bilinearity*: $\forall g, h \in \mathbb{G}$ and $a, b \in_R \mathbb{Z}_N, e(g^a, h^b) = e(g, h)^{ab}$; (2) *Non-degeneracy*: $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$. Assume that the group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map e are computable in polynomial time with respect to a security parameter k, and that the group description of $\mathbb{G}$ and $\mathbb{G}_T$ include the generators of the respective cyclic groups. We denote by $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}, \mathbb{G}_{p_3}$ the subgroups of order p_1, p_2, p_3 in $\mathbb{G}$, respectively.

Definition 6 (*Assumption 1 [18]*)**.** Given a security parameter 1^k, we define the following distribution:

$$(N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^k),$$
$$g_1 \in_R \mathbb{G}_{p_1},\ g_2, X_2, Y_2 \in_R \mathbb{G}_{p_2},\ g_3 \in_R \mathbb{G}_{p_3},$$
$$\alpha, s \in_R \mathbb{Z}_N, \quad D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, g_2, g_3, g_1^{\alpha}X_2, g_1^{s}Y_2),$$
$$T_0 = e(g_1, g_1)^{\alpha s}, \quad T_1 \in_R \mathbb{G}_T.$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is $Adv^1_{\mathcal{A}}(1^k) = |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]|$. We say that $\mathcal{G}$ satisfies Assumption 1 if $Adv^1_{\mathcal{A}}(1^k)$ is negligible for any PPT algorithm $\mathcal{A}$.

Definition 7 (*The General Subgroup Decision Assumption [18]*)**.** Let $Z_0, Z_1, Z_2, \ldots, Z_\phi$ denote a collection of non-empty subsets of $\{1, 2, 3\}$ where each Z_i for $i \geq 2$ satisfies either $Z_0 \cap Z_i \neq \emptyset \neq Z_1 \cap Z_i$ or $Z_0 \cap Z_i = \emptyset = Z_1 \cap Z_i$. Given a security parameter 1^k, we define the following distribution:

$$(N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^k),$$
$$g_{Z_2} \in_R \mathbb{G}_{Z_2}, \ldots, g_{Z_\phi} \in_R \mathbb{G}_{Z_\phi},$$
$$D = (N, \mathbb{G}, \mathbb{G}_T, e, g_{Z_2}, \ldots, g_{Z_\phi}), \quad T_0 \in_R \mathbb{G}_{Z_0},\ T_1 \in_R \mathbb{G}_{Z_1}.$$

Fixing the collection of sets $Z_0, \ldots, Z_\phi$, we define the advantage of an algorithm $\mathcal{A}$ in breaking this assumption to be $Adv^{SD}_{\mathcal{A}}(1^k) = |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]|$. We say that $\mathcal{G}$ satisfies the General Subgroup Decision Assumption if $Adv^{SD}_{\mathcal{A}}(1^k)$ is negligible for any PPT algorithm $\mathcal{A}$ and any suitable collection of subsets $Z_0, \ldots, Z_\phi$.

Remarks. As stated in [18], this assumption already implies that a non-trivial factor of N is hard to be extracted.

Definition 8 (*The Three Party Diffie–Hellman Assumption in a Subgroup [18]*)**.** Given a security parameter 1^k, we define the following distribution:

$$(N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^k),$$
$$g_1 \in_R \mathbb{G}_{p_1},\ g_2 \in_R \mathbb{G}_{p_2},\ g_3 \in_R \mathbb{G}_{p_3},\ x, y, z \in_R \mathbb{Z}_N,$$
$$D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, g_2, g_3, g_2^x, g_2^y, g_2^z),$$
$$T_0 = g_2^{xyz}, \quad T_1 \in_R \mathbb{G}_{p_2}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is $Adv^{3DH}_{\mathcal{A}}(1^k) = |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]|$. We say that $\mathcal{G}$ satisfies The Three Party Diffie–Hellman Assumption in a Subgroup if $Adv^{3DH}_{\mathcal{A}}(1^k)$ is negligible for any PPT algorithm $\mathcal{A}$.

Definition 9 (*The Source Group q-Parallel BDHE Assumption in a Subgroup [18]*)**.** Given a security parameter 1^k and a positive integer q, we define the following distribution:

$$(N = p_1p_2p_3, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^k),$$
$$g_1 \in_R \mathbb{G}_{p_1},\ g_2 \in_R \mathbb{G}_{p_2},\ g_3 \in_R \mathbb{G}_{p_3},$$
$$c, d, f, b_1, \ldots, b_q \in_R \mathbb{Z}_N,$$
$$D = (N, \mathbb{G}, \mathbb{G}_T, e, g_1, g_2, g_3, g_2^f, g_2^{df}, g_2^c, g_2^{c^2}, \ldots,$$
$$g_2^{c^q}, g_2^{c^{q+2}}, \ldots, g_2^{c^{2q}},\ \forall i \in [1, 2q] \setminus \{q+1\},\ j \in [1, q]\ g_2^{c^i/b_j},$$
$$\forall j \in [1, q]\ g_2^{dfb_j};\ \forall i \in [1, q], j, j' \in [1, q] \text{ s.t. } j \neq j'\ g_2^{dfc^ib_{j'}/b_j}),$$
$$T_0 = g_2^{dc^{q+1}}, \quad T_1 \in_R \mathbb{G}_{p_2}.$$

The advantage of an algorithm $\mathcal{A}$ in breaking this assumption is $Adv^q_{\mathcal{A}}(1^k) = |Pr[\mathcal{A}(D, T_0) = 1] - Pr[\mathcal{A}(D, T_1) = 1]|$. We say that $\mathcal{G}$ satisfies The Source Group q-Parallel BDHE Assumption in a Subgroup if $Adv^q_{\mathcal{A}}(1^k)$ is negligible for any PPT algorithm $\mathcal{A}$.

Target collision resistant (TCR) hash function. 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 $Adv^{TCR}_{H,A} = Pr[(x, y) \leftarrow \mathcal{A}(1^k) : H(x) = H(y),\ x \neq y,\ x, y \in DH]$ 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, $Adv^{TCR}_{H,A}$ is negligible.

Strongly existential unforgeable one-time signatures (OTS). An OTS system [36] consists of the following algorithms:

1. $(ssk, svk) \leftarrow OTS.KeyGen(1^k)$: on input a security parameter $k \in \mathbb{N}$, the algorithm outputs a signing/verification key pair (ssk, svk).
2. $\sigma \leftarrow Sign(ssk, M)$: on input ssk and a message $M \in \Gamma_{Sig}$, the algorithm outputs a signature σ, where Γ_{Sig} is the message space of a signature scheme.
3. $1/0 \leftarrow Verify(svk, \sigma, M)$: on input svk, a signature σ and a message M, the algorithm outputs 1 when σ is a valid signature of M, and output 0 otherwise.

An OTS scheme is one-time strongly unforgeable chosen message attack secure if the advantage $Adv^{OTS}_{\mathcal{A}}(1^k)$ is negligible for any PPT adversary $\mathcal{A}$ in the following experiment.

$$Adv^{OTS}_{\mathcal{A}}(1^k) = Pr[Verify(svk, \sigma^*, M^*) = 1 : (ssk, svk)$$
$$\leftarrow OTS.KeyGen(1^k);\ (M, State) \leftarrow \mathcal{A}(svk);$$
$$\sigma \leftarrow Sign(ssk, M);\ (M^*, \sigma^*) \leftarrow \mathcal{A}(svk, \sigma, State);$$
$$(M^*, \sigma^*) \neq (M, \sigma)],$$

where k is the security parameter, $State$ is the state information, $M, M^* \in \Gamma_{Sig}$.

One-time symmetric encryption. A one-time symmetric encryption [35] consists of the following algorithms. Note let $\mathcal{K}_D$ be the key space $\{0, 1\}^{poly(1^k)}$, and SYM be a symmetric encryption scheme, where $poly(1^k)$ is the fixed polynomial size (bound) with respect to the security parameter k. The encryption algorithm $SYM.Enc$ intakes a key $K \in \mathcal{K}_D$ and a message M, outputs a ciphertext C. The decryption algorithm $SYM.Dec$ intakes K and C, outputs M or a symbol $\perp$. The CCA security model for SYM systems is given in [20], we hence omit the details.

4. An adaptively CCA-secure CP-ABPRE

In this section we propose an adaptively CCA-secure CP-ABPRE scheme in the standard model.

4.1. Our approach

We start with Waters-ABE scheme [11], and further extend the scheme to support attribute-based re-encryption. The main technique we use is somewhat similar to the technology of secret sharing. We allow the proxy to use a re-encryption key embedded with the secret key of a delegator to partially "decrypt" an original ciphertext without revealing any information of the underlying plaintext such that only a valid delegatee can recover the plaintext using his/her secret key. Here the most subtle part is the generation of re-encryption key. It can be seen from Definition 3 that the re-encryption key generation algorithm takes the secret key of a delegator and an access structure as input. To preclude the proxy from accessing the underlying plaintext (i.e. the proxy cannot use the re-encryption key to decrypt the original ciphertext directly), we construct the re-encryption key by first using a random factor to mask the secret key of the delegator and next encrypting the factor for the corresponding delegatee. In this case the re-encryption will yield a calculation of the content key (which is used to recover the plaintext) hidden by the random factor which can be only recovered by the delegatee. This method, nevertheless, cannot be secure against collusion attacks. When the proxy colludes with the delegatee, then the entire secret key of the delegator is exposed. To address the problem, we make use of the property of bilinear pairing in the sense that we additionally use random factors (which are unknown to both the proxy and the delegatee) to hide the secret key components of the delegator but these factors will be eliminated eventually in re-encryption phase.

Intuition of adaptive CCA security. As stated in [22], it is challenging to build a PRE (in general) system with CCA security in the standard model. To achieve CCA security in the standard model, we might choose to employ the well-known CHK transformation [37] in our construction. We, however, cannot trivially make use of such a transformation in the PRE cryptographic setting. This is because there is a paradox between the two technologies where PRE allows transformation among different ciphertexts, while the CHK transformation is used to guarantee the integrity of ciphertexts by detecting the modification of ciphertexts. From the literature, we can see that trivially employing the CHK transformation in the context of PRE often yields a Replayable CCA (RCCA) secure [38] system, such as [21,39,40]. Note RCCA security is a weaker notion when compared with CCA security.

Here we tackle the paradox with the help of one-time symmetric encryption system. Our technical roadmap is described as follows. We first sign an original ciphertext in the CHK transformation, and next allow a semi-trusted proxy to hide the re-encryption result in a CCA-secure one-time symmetric encryption system in which the symmetric key is encrypted intended for the delegatee. The delegatee then can recover the underlying plaintext by using his secret key accordingly. It is not difficult to see that by combining the usage of the CHK transformation with a CCA-secure one-time symmetric encryption system we can make the original and re-encrypted ciphertexts hold against CCA. Nonetheless, this method easily leads to a potential problem that the encryption of the symmetric key is vulnerable to be further re-encrypted, i.e. multiple times re-encryption. This conflicts our initial intention of proposing a CP-ABPRE system with single-hop property. To solve this problem, we use a tricky approach in the construction of ciphertext. Due to limited space we refer the reader to Section 4.2 for the details of our construction.

Although the above technology enables our scheme to achieve CCA security, the scheme is only *selectively* CCA secure. This selective security is inherited from the underlying primitive, Waters-ABE scheme, which is proven selectively CPA secure. To convert a selective secure cryptosystem to achieve fully security, we may leverage the dual system encryption technology [17] which is applicable to the context of ABE. Nevertheless, the technology introduced in [17] incurs a limitation that an attribute can be only used once in an access policy. If the attribute is used repeatedly (note the repetition number is limited to some upper bound), the size of system parameters will be enlarged. This does not scale well in practice due to its loss of efficiency and expressiveness.

Fortunately, Lewko and Waters [18] introduce a new approach to achieve full security without any loss of efficiency in the ABE cryptographic setting. This new technique requires a combination of dual system encryption technology and selective proof technique. We mainly follow the framework of [18] to achieve fully security. Generally, we use the elements of subgroup $\mathbb{G}_{p_1}$ to represent the original components of (extended) Waters-ABE system and additionally, use the elements of subgroup $\mathbb{G}_{p_3}$ to randomize users' secret keys. We use the elements of subgroup $\mathbb{G}_{p_2}$ in our security proof only. Note we will further introduce our security proof framework in Section 5.

4.2. A basic construction

The description of our new CP-ABPRE scheme with adaptive CCA security is as follows. Unless stated otherwise, we let $\mathcal{U}$ be an attribute universe, and S be an attribute set, $S \subseteq \mathcal{U}$.

1. **Setup**$(1^k, \mathcal{U})$. The setup algorithm runs $(N, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathcal{G}(1^k)$, where $N = p_1p_2p_3$ is the order of group $\mathbb{G}$ and p_1, p_2, p_3 are distinct primes. Let $\mathbb{G}_{p_i}$ denote the subgroup of order p_i in group $\mathbb{G}$. It chooses $a, \alpha, \kappa, \beta, \epsilon \in_R \mathbb{Z}_N$, $g, \hat{g}_1 \in_R \mathbb{G}_{p_1}$, two TCR hash functions $TCR_1 : \mathbb{G}_T \to \mathbb{Z}_N$, $TCR_2 : \mathbb{G}_T \to \{0,1\}^{poly(1^k)}$, a CCA-secure one-time symmetric encryption system *SYM* and a strongly existential unforgeable one-time signature system *OTS*. For each attribute $i \in \mathcal{U}$, it chooses $h_i \in_R \mathbb{Z}_N$. The public parameters *param* are $(N, g, \hat{g}_1, g^a, g^\kappa, g^\beta, g^\epsilon, e(g,g)^\alpha, \forall i \in \mathcal{U}\ H_i = g^{h_i}, TCR_1, TCR_2, SYM, OTS)$, and the master secret key *msk* is (g^α, g_3), where g_3 is a generator of $\mathbb{G}_{p_3}$.
2. **KeyGen**$(param, msk, S)$. The key generation algorithm chooses $t, u \in_R \mathbb{Z}_N$, and $R, R', R'', \{R_i\}_{i\in S} \in_R \mathbb{G}_{p_3}$. The secret key sk_S is
$$(S, K = g^\alpha g^{at} g^{\kappa u} R, K' = g^u R', K'' = g^t R'', \forall i \in S\ K_i = H_i^t R_i).$$
3. **Encrypt**$(param, (A, \rho), m)$. Given an LSSS access structure (A, ρ) and a message $m \in \mathbb{G}_T$ in which A is an $l \times n$ matrix and ρ is a map from each row A_j to an attribute $\rho(j)$, the encryption algorithm works as follows.
   (a) Choose a random vector $v = (s, v_2, \ldots, v_n) \in_R \mathbb{Z}_N^n$.
   (b) For each row A_j, choose $r_j \in_R \mathbb{Z}_N$, run $(ssk, svk) \leftarrow OTS.KeyGen(1^k)$ and set
   $B_0 = m \cdot e(g,g)^{\alpha s}$, $B_1 = g^s, B_2 = (g^\kappa)^s$,
   $B_3 = (\hat{g}_1^{svk} g^\beta)^s$, $B_4 = (g^\epsilon)^s$,
   $\forall j \in [1, l](C_j = (g^a)^{A_j v} H_{\rho(j)}{}^{-r_j}, D_j = g^{r_j})$,
   $E = Sign(ssk, (B_0, B_1, B_3, \forall j \in [1,l]\ (C_j, D_j)))$.
   (c) Output the original ciphertext $C = (svk, B_0, B_1, B_2, B_3, B_4, \forall j \in [1,l]\ (C_j, D_j), E)$. Note $\{\rho(j) | 1 \le j \le l\}$ are the attributes used in (A, ρ).
4. **ReKeyGen**$(param, sk_S, (A', \rho'))$. Given a secret key $sk_S = (S, K, K', K'', \forall i \in S\ K_i)$ and an LSSS access structure (A', ρ'), the re-encryption key $rk_{S\to(A',\rho')}$ is generated as follows.
   (a) Choose $\theta_1, \theta_2, \theta_3 \in_R \mathbb{Z}_N$, $\delta \in_R \mathbb{G}_T$, set $rk_1 = (Kg^{\kappa\theta_1} g^{a\theta_2})^{TCR_1(\delta)} g^{\epsilon\theta_3}$, $rk_2 = (K' g^{\theta_1})^{TCR_1(\delta)}$, $rk_3 = (K'' g^{\theta_2})^{TCR_1(\delta)}$, $rk_4 = g^{\theta_3}$, $\forall i \in S\ rk_{5,i} = (K_i H_i^{\theta_2})^{TCR_1(\delta)}$.

(b) Choose a random vector $v^{(rk)} = (s^{(rk)}, v_2^{(rk)}, \ldots, v_n^{(rk)}) \in_R \mathbb{Z}_N^n$. For each row A'_j of A', choose $r_j^{(rk)} \in_R \mathbb{Z}_N$, run $(ssk^{(rk)}, svk^{(rk)}) \leftarrow OTS.KeyGen(1^k)$ and set rk_6 as

$svk^{(rk)}$, $B_0^{(rk)} = \delta \cdot e(g,g)^{\alpha s^{(rk)}}$, $B_1^{(rk)} = g^{s^{(rk)}}$,

$B_2^{(rk)} = (g^\kappa)^{s^{(rk)}}$, $B_3^{(rk)} = (\hat{g}_1^{svk^{(rk)}} g^\beta)^{s^{(rk)}}$,

$\forall j \in [1,l]\ (C_j^{(rk)} = (g^a)^{A'_j v^{(rk)}} H_{\rho'(j)}^{-r_j^{(rk)}}, D_j^{(rk)} = g^{r_j^{(rk)}})$,

$E^{(rk)} = Sign(ssk^{(rk)}, (B_0^{(rk)}, B_1^{(rk)}, B_3^{(rk)},$

$\forall j \in [1,l]\ (C_j^{(rk)}, D_j^{(rk)})))$.

(c) Output the re-encryption key $rk_{S\to(A',\rho')} = (rk_1, rk_2, rk_3, rk_4, \forall i \in S\ rk_{5,i}, rk_6)$.

5. **ReEnc**$(param, rk_{S\to(A',\rho')}, C)$. Parse the original ciphertext C under (A,ρ) as $(svk, B_0, B_1, B_2, B_3, B_4, \forall j \in [1,l]\ (C_j, D_j), E)$, and the re-encryption key $rk_{S\to(A',\rho')}$ as $(rk_1, rk_2, rk_3, rk_4, \forall i \in S\ rk_{5,i}, rk_6)$.

(a) Check the validity of the original ciphertext C as

$$e(B_1, g^\kappa) \stackrel{?}{=} e(B_2, g),$$
$$e(B_1, \hat{g}_1^{svk} g^\beta) \stackrel{?}{=} e(B_3, g),$$
$$e(B_1, g^\epsilon) \stackrel{?}{=} e(B_4, g),$$
$$e\left(\prod_{\rho(j)\in S} C_j^{w_j}, g\right) \stackrel{?}{=} e(B_1, g^a) \times \prod_{\rho(j)\in S} (e(D_j^{-1}, H_{\rho(j)}^{w_j})), \quad S \stackrel{?}{\models} (A,\rho)$$

$$Verify(svk, (E, (B_0, B_1, B_3, \forall j \in [1,l]\ (C_j, D_j)))) \stackrel{?}{=} 1, \quad (1)$$

where w_j are the constants chosen by the proxy such that the following holds $\sum_{\rho(j)\in S} w_j A_j = (1,0,\ldots,0)$. If Eq. (1) does not hold, output $\perp$. Otherwise, proceed.

(b) Compute $F = \frac{e(B_1,rk_1)e(B_2,rk_2)^{-1}e(B_4,rk_4)^{-1}}{(\prod_{\rho(j)\in S}(e(C_j,rk_3)e(D_j,rk_{5,j}))^{w_j})}$, and further run $\sigma_1 = SYM.Enc(TCR_2(key), G)$, where $G = (C \| rk_6 \| F)$ and $key \in_R \mathbb{G}_T$.

(c) Choose a random vector $v^{(re)} = (s^{(re)}, v_2^{(re)}, \ldots, v_n^{(re)}) \in_R \mathbb{Z}_N^n$. For each row A'_j of A', choose $r_j^{(re)} \in_R \mathbb{Z}_N$, run $(ssk^{(re)}, svk^{(re)}) \leftarrow OTS.KeyGen(1^k)$ and set σ_2 as

$svk^{(re)}$, $B_0^{(re)} = key \cdot e(g,g)^{\alpha s^{(re)}}$, $B_1^{(re)} = g^{s^{(re)}}$,

$B_2^{(re)} = (g^\kappa)^{s^{(re)}}$, $B_3^{(re)} = (\hat{g}_1^{svk^{(re)}} g^\beta)^{s^{(re)}}$,

$\forall j \in [1,l]\ (C_j^{(re)} = (g^a)^{A'_j v^{(re)}} H_{\rho'(j)}^{-r_j^{(re)}}, D_j^{(re)} = g^{r_j^{(re)}})$,

$E^{(re)} = Sign(ssk^{(re)}, (B_0^{(re)}, B_1^{(re)}, B_3^{(re)},$

$\forall j \in [1,l]\ (C_j^{(re)}, D_j^{(re)})))$.

(d) Output the re-encrypted ciphertext $C_R = (\sigma_1, \sigma_2)$ under (A', ρ').

6. **Dec**$(param, sk_S, C)$. Parse the original ciphertext C under (A,ρ) as $(svk, B_0, B_1, B_2, B_3, B_4, \forall j \in [1,l]\ (C_j, D_j), E)$, and the secret key sk_S as $(S, K, K', K'', \forall i \in S\ K_i)$. The decryption algorithm chooses a set of constants $w_j \in_R \mathbb{Z}_N$ such that $\sum_{\rho(j)\in S} w_j A_j = (1,0,\ldots,0)$, and next recovers the message as follows.

(a) If Eq. (1) does not hold, output $\perp$. Otherwise, proceed.

(b) Compute $e(B_1,K)e(B_2,K')^{-1}/(\prod_{\rho(j)\in S}(e(C_j,K'')e(D_j,K_{\rho(j)}))^{w_j}) = e(g,g)^{\alpha s}$, and output the message $m = B_0/e(g,g)^{\alpha s}$.

7. **Dec**$_R(param, sk_S, C_R)$. Parse the re-encrypted ciphertext C_R under (A',ρ') as (σ_1, σ_2), and the secret key sk_S as $(S, K, K', K'', \forall i \in S\ K_i)$.

(a) Check the validity of σ_2 as

$$e(B_1^{(re)}, g^\kappa) \stackrel{?}{=} e(B_2^{(re)}, g),$$
$$e(B_1^{(re)}, \hat{g}_1^{svk^{(re)}} g^\beta) \stackrel{?}{=} e(B_3^{(re)}, g), \quad S \stackrel{?}{\models} (A',\rho'),$$
$$e\left(\prod_{\rho'(j)\in S} (C_j^{(re)})^{w_j^{(re)}}, g\right) \stackrel{?}{=} e(B_1^{(re)}, g^a) \cdot \prod_{\rho'(j)\in S} (e((D_j^{(re)})^{-1}, H_{\rho'(j)}^{w_j^{(re)}})),$$

$$Verify(svk^{(re)}, (E^{(re)}, (B_0^{(re)}, B_1^{(re)}, B_3^{(re)}, \forall j \in [1,l]\ (C_j^{(re)}, D_j^{(re)})))) \stackrel{?}{=} 1, \quad (2)$$

where $w_j^{(re)}$ are the constants chosen by the decryptor such that $\sum_{\rho'(j)\in S} w_j^{(re)} A'_j = (1,0,\ldots,0)$. If Eq. (2) does not hold, output $\perp$. Otherwise, proceed.

(b) Compute $e(B_1^{(re)}, K)e(B_2^{(re)}, K')^{-1}/(\prod_{\rho'(j)\in S}(e(C_j^{(re)}, K'')e(D_j^{(re)}, K_{\rho'(j)}))^{w_j^{(re)}}) = e(g,g)^{\alpha s^{(re)}}$, and output $key = B_0^{(re)}/e(g,g)^{\alpha s^{(re)}}$.

(c) Run $G = SYM.Dec(TCR_2(key), \sigma_1)$.

(d) Parse G as (C, rk_6, F). If either Eq. (1) or the following verification for rk_6 does not hold, output $\perp$. Otherwise, proceed.

$$e(B_1^{(rk)}, g^\kappa) \stackrel{?}{=} e(B_2^{(rk)}, g),$$
$$e(B_1^{(rk)}, \hat{g}_1^{svk^{(rk)}} g^\beta) \stackrel{?}{=} e(B_3^{(rk)}, g),$$
$$e\left(\prod_{\rho'(j)\in S} (C_j^{(rk)})^{w_j^{(rk)}}, g\right) \stackrel{?}{=} e(B_1^{(rk)}, g^a) \cdot \prod_{\rho'(j)\in S} (e((D_j^{(rk)})^{-1}, H_{\rho'(j)}^{w_j^{(rk)}})),$$

$$Verify(svk^{(rk)}, (E^{(rk)}, (B_0^{(rk)}, B_1^{(rk)}, B_3^{(rk)}, \forall j \in [1,l]\ (C_j^{(rk)}, D_j^{(rk)})))) \stackrel{?}{=} 1, \quad S \stackrel{?}{\models} (A',\rho'), \quad (3)$$

where $w_j^{(rk)}$ are the constants chosen by the decryptor such that $\sum_{\rho'(j)\in S} w_j^{(rk)} A'_j = (1,0,\ldots,0)$.

(e) Compute $e(B_1^{(rk)}, K)e(B_2^{(rk)}, K')^{-1}/(\prod_{\rho'(j)\in S}(e(C_j^{(rk)}, K'')e(D_j^{(rk)}, K_{\rho'(j)}))^{w_j^{(rk)}}) = e(g,g)^{\alpha s^{(rk)}}$, and then $B_0^{(rk)}/e(g,g)^{\alpha s^{(rk)}} = \delta$. Compute $F^{TCR_1(\delta)^{-1}} = e(g,g)^{\alpha s}$, and finally output $m = B_0/e(g,g)^{\alpha s}$.

- **Correctness for original ciphertext**.

$$\begin{aligned} & e(B_1,K)e(B_2,K')^{-1} \Big/ \prod_{\rho(j)\in S} (e(C_j,K'')e(D_j,K_{\rho(j)}))^{w_j} \\ &= \frac{e(g^s, g^\alpha g^{at} g^{\kappa u} R) e(g^{\kappa s}, g^u R')^{-1}}{\prod_{\rho(j)\in S} (e(g^{aA_j v} H_{\rho(j)}^{-r_j}, g^t R'') e(g^{r_j}, H_{\rho(j)}^t R_{\rho(j)}))^{w_j}} \\ &= \frac{e(g^s, g^\alpha g^{at})}{e(g, g^{at})^{\sum_{\rho(j)\in S} A_j v w_j}} \\ &= e(g^s, g^\alpha). \end{aligned}$$

Hence, we have $B_0/e(g^s,g^\alpha) = m \cdot e(g,g)^{\alpha s}/e(g^s,g^\alpha) = m$.

Table 2
Complexity comparison with [32].

Schemes	Phases	Communication	Computation
[32]	*ReKeyGen*	$O(a)\|\mathbb{G}_{p_1p_3}\|+O(l)\|\mathbb{G}_{p_1}\|+O(1)\|\mathbb{G}_T\|$	$O(l)c_e + O(a)c_e$
	ReEnc	$O(1)\|SYM\|+O(l)\|\mathbb{G}_{p_1}\|+O(1)\|\mathbb{G}_T\|$	$O(a)c_p + O(l)c_e + O(a)c_e$
Improved version	*ReKeyGen*	$O(1)\|\mathbb{G}_{p_1p_3}\|+O(l)\|\mathbb{G}_{p_1}\|+O(1)\|\mathbb{G}_T\|$	$O(l)c_e$
	ReEnc	$O(1)\|SYM\|+O(l)\|\mathbb{G}_{p_1}\|+O(1)\|\mathbb{G}_T\|$	$O(1)c_p + O(l)c_e$

- **Correctness for re-encrypted ciphertext**.

$$F = \frac{e(B_1, rk_1)e(B_2, rk_2)^{-1}e(B_4, rk_4)^{-1}}{\left(\prod_{\rho(j)\in S}(e(C_j, rk_3)e(D_j, rk_{4,j}))^{w_j}\right)}$$
$$= \frac{e(g^s, g^{\alpha}g^{at}g^{a\theta_2})^{TCR_1(\delta)}}{e(g, g^{atTCR_1(\delta)})^{\sum_{\rho(j)\in S} A_j w_j v}\, e(g, g^{a\theta_2 TCR_1(\delta)})^{\sum_{\rho(j)\in S} A_j w_j v}}$$
$$= e(g, g)^{\alpha s TCR_1(\delta)}.$$

Thus we have $F^{TCR_1(\delta)^{-1}} = (e(g, g)^{\alpha s TCR_1(\delta)})^{TCR_1(\delta)^{-1}} = e(g, g)^{\alpha s}$, and $B_0/e(g, g)^{\alpha s} = m$.

4.3. Discussions

1. Non-interaction re-encryption key generation. From the construction of re-encryption key, we can see that it only needs the secret key of a delegator but not that of the corresponding delegatee. In other words, it requires the delegator to participate into the re-encryption key generation phase. Accordingly the system does not require any interaction between the delegator and the delegatee that saves the bandwidth of communication.
2. Unidirectional property. Besides, the scheme is unidirectional as the proxy cannot leverage the "same" re-encryption key to make ciphertexts transformation from delegator to delegatee as well as the other way round.
3. Single-hop property. As to the single-hop property, we can see that a re-encrypted ciphertext consists of a symmetric encryption σ_1 (which cannot be further converted) and an asymmetric encryption σ_2. In the construction we let σ_2 exclude the component $B_4^{(re)}$, which is a crucial component for re-encryption, such that it is impossible to make σ_2 be further re-encrypted.
4. Collusion resistance. Furthermore, the proxy cannot compromise the secret key of a delegator without knowledge of $\theta_1, \theta_2, \theta_3$ even colluding with the corresponding delegatee. This allows our system to achieve collusion resistance.

4.4. Efficiency improvement

From our basic construction, it can be seen that the size of re-encryption key is linearly in both the number of attributes associated with the secret key of the delegator and the size of access policy associated with an encryption of a random hiding factor for the delegatee. Moreover, the cost of generating F in the algorithm *ReEnc* suffers from $O(\hat{a})$ parings complexity, where $\hat{a}$ is the number of attributes sharing between the access structure (A, ρ) and the attribute set S. Below we are going to improve the communication and computation complexity of our system in the re-encryption key generation and re-encryption phases.

A Naive solution. To reduce the size of re-encryption key and the re-encryption complexity, we may directly leverage the following solution. We off-load the re-encryption task to the side of PKG. Namely, the PKG will take place of the proxy to fulfill the re-encryption. Here we modify the algorithm *ReEnc* as: after verifying the validity of the original ciphertext by Eq. (1), the PKG computes $F = e(B_1, g^{\alpha}\cdot g^{TCR_1(\zeta)}) = e(g^s, g^{\alpha}\cdot g^{TCR_1(\zeta)})$, and rk_6, an encryption of ζ under (A', ρ') generated as in the original algorithm *ReKeyGen*, where $\zeta \in_R \mathbb{G}_T$, g^{α} is one of the components of *msk* which is known by the PKG. The PKG then computes σ_1, G, and σ_2 as in the original algorithm *ReEnc*. It is not difficult to see that if the attribute set of a re-encrypted ciphertext receiver matches (A', ρ'), the receiver can recover ζ by using his/her secret key, and further gain access to the underlying plaintext with knowledge of $e(g^s, g^{\alpha})$. However, this solution contradicts our initial idea of proxy re-encryption whereby we assume the proxy (taking charge of re-encryption) is a semi-trusted party only. The solution here indicates that the proxy, i.e. the PKG, is fully trusted. If the proxy can be fully trusted, why we do not directly allow it to gain access to the data and further construct a new encryption (of the data) for corresponding receivers?

A better solution. Here we present a better solution to reduce the complexity of our basic construction without contradicting the purpose of proxy re-encryption. We delegate the re-encryption key generation to the PKG. In the algorithm *ReKeyGen*, the PKG sets the re-encryption key as $rk_1 = g^{\alpha}\cdot g^{TCR_1(\eta)}\cdot R\cdot g^{\epsilon\theta_3}$, $rk_2 = g^{\theta_3}$, rk_3, an encryption of η under (A', ρ'), is generated as rk_6 in the original algorithm *ReKeyGen*, where $\eta \in_R \mathbb{G}_T$, and $\theta_3 \in_R \mathbb{Z}_N$. In the algorithm *ReEnc*, the proxy computes F as $e(B_1, rk_1)\cdot e(B_4, rk_2)^{-1} = e(g^s, g^{\alpha}\cdot g^{TCR_1(\eta)}\cdot R\cdot g^{\epsilon\theta_3})\cdot e(g^{\epsilon s}, g^{\theta_3})^{-1} = e(g^s, g^{\alpha}\cdot g^{TCR_1(\eta)})$, and next computes σ_1, G, and σ_2 as in the original algorithm *ReEnc*. It is easy to see that if η encrypted under a new access policy (A', ρ') can be retrieved, then the message m is recovered.

Improvement analysis. Below we show how much complexity improvement we make by using the solution above. A comparison between our basic construction and the one improved by using the new solution in terms of re-encryption key generation and re-encryption phases is shown in Table 2. Note we let $|\mathbb{G}_{p_1}|$, $|\mathbb{G}_{p_1p_3}|$ denote the bit-length of an element in $\mathbb{G}_{p_1}$ and $\mathbb{G}_{p_1p_3}$, $|\mathbb{G}_T|$ denote the bit-length of an element in $\mathbb{G}_T$ and $|SYM|$ denote the bit-length of a symmetric encryption.

- In re-encryption key generation phase: the improved scheme saves $O(a)|\mathbb{G}_{p_1p_3}|$ and $O(a)c_e$ in terms of communication and computation cost, respectively.
- In re-encryption phase: the improved scheme reduces the exponentiation cost to $O(l)c_e$, and lessens the pairings computation cost from $O(a)c_p$ to constant.

5. Security analysis

Before presenting the formal security analysis, we give some intuition as to why our system is secure against CCA. For the security of original ciphertext, we let $C^* = (svk^*, B_0^*, B_1^*, B_2^*, B_3^*, B_4^*, \forall j \in [1, l](C_j^*, D_j^*), E^*)$ be the challenge ciphertext of m_b. Suppose $\mathcal{A}$ following the constraints defined in Definition 4 will try to guess the value of the bit b by leveraging the responses of $\mathcal{O}_{re}$ and $\mathcal{O}_{dec}$. Generally, we can see that C^* consists of an encryption of a CPA-secure Waters-ABE system and a one-time signature. If $\mathcal{A}$ mutates C^*, and next submits the resulting ciphertext to the oracles, then the mutation can be noticed with non-negligible probability. This is so because $B_0^*, B_1^*, B_3^*, \forall j \in [1, l](C_j^*, D_j^*)$ are signed in E^* (assuming *OTS* is a strongly existential unforgeable one-time signature system),

and the integrity of B_2^* and B_4^* is bound by B_1^* (which can be seen from Eq. (1)). If C^* is mutated, then Eq. (1) will not hold. Therefore, the value of the bit b is hidden from $\mathcal{A}$ (i.e. $\mathcal{A}$ will not obtain additional advantage with the help of $\mathcal{O}_{re}$ and $\mathcal{O}_{dec}$ in winning the game).

For the security of re-encrypted ciphertext, we let $C_R^* = (\sigma_1^*, \sigma_2^*)$ be the ciphertext of m_b. Following the restrictions of Definition 5, $\mathcal{A}$ will try to win the game by leveraging the responses of $\mathcal{O}_{dec_R}$. Note in the game of re-encrypted ciphertext $\mathcal{A}$ is not given $\mathcal{O}_{re}$ as it is allowed to obtain any re-encryption key. Recall that a re-encrypted ciphertext cannot be further transformed, thus $\mathcal{A}$ does not gain additional advantage even given any re-encryption key here. From C_R^*, we can see that σ_1^* is an CCA-secure one-time symmetric encryption, and the structure of σ_2^* looks somewhat "identical" to that of the original ciphertext. Similarly, $\sigma_2^* = (svk^{(re)}, B_0^{(re)}, B_1^{(re)}, B_2^{(re)}, B_3^{(re)}, \forall j \in [1, l](C_j^{(re)}, D_j^{(re)}), E^{(re)})$ can also be regarded as a combination of an encryption of a CPA-secure Waters-ABE system and a one-time signature such that the oracle can tell any mutation for σ_2^* with non-negligible probability. Thus the value of the bit b is hidden from $\mathcal{A}$ as well.

Therefore we have the following theorem.

Theorem 1. *Suppose Assumption* 1*, the general subgroup decision assumption, the three party Diffie–Hellman assumption in a subgroup, and the source q-parallel BDHE assumption in a subgroup hold, SYM is a CCA-secure one-time symmetric encryption, OTS is a strongly existential unforgeable one-time signature, and TCR_1, TCR_2 are the TCR hash functions, our system is IND-CCA secure in the standard model.*

We prove our scheme by following the security framework of [18]. Before introducing our game setting, we define the semi-functional keys and semi-functional ciphertexts which are used in our proof. We let g_2 denote a generator of the subgroup $\mathbb{G}_{p_2}$.

Semi-functional ciphertexts. We generate the normalciphertext components $svk, B_0, B_1, B_2, B_3, B_4, \forall j \in [1, l]\ (C_j, D_j)$. We next choose $a', \kappa', s', \beta', \beta'', \epsilon' \in_R \mathbb{Z}_N$, a vector $\hat{v} = (s', v_2', \ldots, v_n') \in_R \mathbb{Z}_N^n$, for each attribute i choose $\xi_i \in_R \mathbb{Z}_N$, for each $j \in [1, l]$ choose $\gamma_j \in_R \mathbb{Z}_N$ and set the ciphertext as:

$$svk,\ B_0' = B_0, \qquad B_1' = B_1 g_2^{s'}, \qquad B_2' = B_2 g_2^{\kappa' s'},$$

$$B_3' = B_3 (g_2^{\beta'' svk} g_2^{\beta'})^{s'},$$

$$B_4' = B_4 (g_2^{\epsilon'})^{s'}, \quad \forall j \in [1, l]\ (C_j' = C_j g_2^{a' A_j \hat{v}} g_2^{-\xi_{\rho(j)} \gamma_j}, D_j' = D_j g_2^{\gamma_j}),$$

$$E' = Sign(ssk, (B_0', B_1', B_3', \forall j \in [1, l]\ (C_j', D_j'))).$$

The structure of the elements in $\mathbb{G}_{p_2}$ is the parallel copy of that of the elements in $\mathbb{G}_{p_1}$. a', κ', ξ_i are common values to the nominal and temporary semi-functional keys. Note the above ciphertext is the original semi-functional ciphertext. The construction of the re-encrypted semi-functional ciphertext is identical to that of the original semi-functional ciphertext except that σ_1 is the symmetric encryption for G (note σ_2 is the encryption of the symmetric key *key*).

Semi-functional keys. By following [18], we set the semi-functional key as follows.

$$KW, K', K'', \quad \forall i \in S\ K_i,$$

where $K, K', K'', K_{i\{i \in S\}}$ are the normal key components generated by calling the algorithm *KeyGen*, and $W \in_R \mathbb{G}_{p_2}$.

To adapt the security framework of [18] to our system, we need two additional types of semi-functional keys which are defined as follows.

Nominal semi-functional keys. The nominal semi-functional keys are set as:

$$Kg_2^{a' t' + \kappa' u'}, K' g_2^{u'}, K'' g_2^{t'}, \quad \forall i \in S\ K_i g_2^{t' \xi_i},$$

where $K, K', K'', K_{i\{i \in S\}}$ are the normal key components, and $t', u' \in_R \mathbb{Z}_N$.

Temporary semi-functional keys. The temporary semi-functional keys are set as:

$$KW, K' g_2^{u'}, K'' g_2^{t'}, \quad \forall i \in S\ K_i g_2^{t' \xi_i},$$

where K, K', K'', K_i are the normal key components, $W \in_R \mathbb{G}_{p_2}$ and $t', u' \in_R \mathbb{Z}_N$.

We will prove Theorem 1 in a hybrid argument over a sequence of games. In all the games (to be defined) $\mathcal{C}$ will play with $\mathcal{A}$, and the total number of queries is $q = q_{sk} + q_{rk} + q_{re} + q_{dec}$, where $q_{sk}, q_{rk}, q_{re}, q_{dec}$ denote the number of the secret key extraction, re-encryption key extraction, re-encryption and decryption queries, respectively. We define $Game_{real}$ to be the first game. It is the IND-CCA security game for CP-ABPRE systems in which the challenge ciphertext (note by the challenge ciphertext we mean both original ciphertext and re-encrypted ciphertext) is normal. In this game, $\mathcal{C}$ will use normal secret keys to respond secret key extraction queries. Besides, $\mathcal{C}$ will first generate normal secret keys, and next leverage these keys to respond the re-encryption key extraction, re-encryption and decryption queries, namely, the re-encryption keys, re-encryption results and decryption results are indirectly computed from the normal secret keys. We define $Game_0$ to be the second game which is identical to $Game_{real}$ except that the challenge ciphertext is semi-functional. Hereafter by "keys" (resp. "key") we mean the secret key(s) (constructed by $\mathcal{C}$) used to respond the secret key extraction, re-encryption key extraction, re-encryption and decryption queries. In the following games, we will convert the "keys" to be semi-functional one by one. But for clarity we make the conversion in such a way that we first turn the "keys" for the secret key extraction queries, and then convert the "keys" for the re-encryption key extraction queries, the re-encryption queries and the decryption queries in sequence. Besides, $\mathcal{A}$ is only allowed to issue one query in each of the following games. We further define $Game_i$ as follows, where $i \in [1, q]$. We let $j_\tau \in [1, q_\tau]$, where $\tau \in \{sk, rk, re, dec\}$. For each game $Game_{j_\tau}$ we define two sub-games $Game_{j_\tau}^N$ and $Game_{j_\tau}^T$ in which the challenge ciphertext is semi-functional. In $Game_{j_\tau}^N$ the first $(j-1)_\tau$ "keys" are semi-functional, the j_τ-th "key" is nominal semi-functional, and the rest of "keys" are normal. In $Game_{j_\tau}^T$ the first $(j-1)_\tau$ "keys" are semi-functional, the j_τ-th "key" is temporary semi-functional, and the remaining "keys" are normal. To transform $Game_{(j-1)_\tau}$ (where j_τ-th "key" is normal) to $Game_{j_\tau}$ (where j_τ-th "key" is semi-functional), we start from converting $Game_{(j-1)_\tau}$ to $Game_{j_\tau}^N$, then to $Game_{j_\tau}^T$, and finally to $Game_{j_\tau}$. Note to get from $Game_{j_\tau}^N$ to $Game_{j_\tau}^T$, we treat the simulations for the queries of Phase 1 and that of Phase 2 differently: the former is based on the three party Diffie–Hellman assumption, and the latter is based on the source group q-parallel BDHE assumption. We will show the proof details soon. In $Game_q = Game_{q_{dec}}$ all "keys" are semi-functional, and the challenge ciphertext is semi-functional for one of the given messages. We define $Game_{final}$ to be the final game in which all "keys" are semi-functional and the challenge ciphertext is semi-functional for a random message, independent of the two message given by $\mathcal{A}$. In the last game, the advantage of $\mathcal{A}$ will be 0. We will prove the above games to be indistinguishable by the following lemmas. Note below we implicitly assume *SYM* is a CCA-secure one-time symmetric encryption, *OTS* is a strongly existential unforgeable one-time signature, TCR_1, TCR_2 are TCR hash functions and it is hard to find a non-trivial factor of N (for Lemmas 3 and 4).

Lemma 1. *If there is a PPT algorithm $\mathcal{A}$ such that $Game_{real}$ $Adv_{\mathcal{A}}^{CP\text{-}ABPRE} - Game_0 Adv_{\mathcal{A}}^{CP\text{-}ABPRE} = \varphi$, we can build a reduction algorithm $\mathcal{C}$ that breaks the general subgroup decision assumption with advantage φ.*

Proof. $\mathcal{C}$ is going to break the general subgroup decision assumption with sets $Z_0 = \{1\}, Z_1 = \{1, 2\}, Z_2 = \{1\}, Z_3 = \{3\}$ by using $\mathcal{A}$ whom can notice the difference between $Game_{real}$ and $Game_0$ with advantage φ. $\mathcal{C}$ is given g_1, g_3, T, where g_1, g_3 are the generators of $\mathbb{G}_{p_1}, \mathbb{G}_{p_3}$, and T is either a random value of $\mathbb{G}_{p_1}$ or a random value of $\mathbb{G}_{p_1p_2}$. $\mathcal{C}$ chooses $\alpha, a, \kappa, \beta, \hat{\beta}, \epsilon, \{h_i\} \in_R \mathbb{Z}_N$, and sets $param = (N, g = g_1, \hat{g}_1 = g_1^{\hat{\beta}}, g^a = g_1^a, g^\kappa = g_1^\kappa, g^\beta = g_1^\beta, g^\epsilon = g_1^\epsilon, e(g, g)^\alpha = e(g_1, g_1)^\alpha, \forall i \in \mathcal{U}\ H_i = g^{h_i} = g_1^{h_i}, TCR_1, TCR_2, SYM, OTS)$, $msk = (g^\alpha, g_3)$, where $\mathcal{C}$ chooses TCR_1, TCR_2, SYM, OTS as in the real scheme.

For secret key extraction queries, $\mathcal{C}$ responds normal secret keys to $\mathcal{A}$ by calling the algorithm *KeyGen* (with knowledge of *msk*). Besides, $\mathcal{C}$ will construct the corresponding normal secret keys, and next leverage these keys to respond other kinds of queries.

In the challenge phase, for the security of original ciphertext, $\mathcal{A}$ outputs $(A^*, \rho^*), m_0, m_1$ to $\mathcal{C}$. $\mathcal{C}$ chooses a vector $\bar{v} = (1, \bar{v}_2, \ldots, \bar{v}_n) \in_R \mathbb{Z}_N^n$, $\forall j \in [1, l]\ \bar{r}_j \in_R \mathbb{Z}_N$, and $b \in \{0, 1\}$. It implicitly sets g^s to be the $\mathbb{G}_{p_1}$ part of T, $v = s\bar{v}$ and $r_j = s\bar{r}_j$. It then sets the challenge original ciphertext as:

$$B'_0 = m_b \cdot e(g_1, T)^\alpha, \quad B'_1 = T, \quad B'_2 = T^\kappa, \quad B'_3 = T^{\hat{\beta}svk+\beta},$$
$$B'_4 = T^\epsilon, \quad \forall j \in [1, l] \quad C'_j = T^{aA_j^*\bar{v}} T^{-\bar{r}_j h_{\rho^*(j)}}, \quad D'_j = T^{\bar{r}_j},$$
$$E' = Sign(ssk, (B'_0, B'_1, B'_3, \forall j \in [1, l]\ (C'_j, D'_j))),$$

where $(ssk, svk) \leftarrow OTS.KeyGen(1^k)$, and we suppose each svk and E' are identically distributed in their corresponding space of *OTS*. For the security of re-encrypted ciphertext, $\mathcal{A}$ outputs (A^*, ρ^*), (A, ρ) and m_0, m_1 to $\mathcal{C}$. $\mathcal{C}$ first generates a re-encryption key $rk_{S\to(A^*,\rho^*)}$ by using a normal secret key sk_S and constructs an original ciphertext for m_b under (A, ρ) by calling algorithm *Encrypt*, where $S \models (A, \rho)$. $\mathcal{C}$ then re-encrypts the original ciphertext, and constructs σ_1 as in the real scheme. It then constructs σ_2 as the original ciphertext (excluding B'_4), and outputs the re-encrypted ciphertext (σ_1, σ_2).

If $T \in \mathbb{G}_{p_1}$, this game is in $Game_{real}$. If $T \in \mathbb{G}_{p_1p_2}$, this game is in $Game_0$. This implicitly sets $g_2^{s'}$ to be the $\mathbb{G}_{p_2}$ part of T, $\kappa' = \kappa$ modulo p_2, $a' = a$ modulo p_2, $\beta'' = \hat{\beta}$ modulo p_2, $\beta' = \beta$ modulo p_2, $\epsilon' = \epsilon$ modulo p_2, $\hat{v} = s'\bar{v}$ modulo p_2, and for each j, $\xi_{\rho^*(j)} = h_{\rho^*(j)}$ modulo p_2, and $\gamma_j = s'\bar{r}_j$ modulo p_2. Note the above values are properly distributed. Thus if $\mathcal{A}$ can tell the difference between $Game_{real}$ and $Game_0$ with advantage φ, $\mathcal{C}$ can use $\mathcal{A}$ to break the general subgroup decision assumption with advantage φ. □

Lemma 2. *If there is a PPT algorithm $\mathcal{A}$ such that $Game_{(j-1)_\tau} Adv_{\mathcal{A}}^{CP\text{-}ABPRE} - Game_{j_\tau}^N Adv_{\mathcal{A}}^{CP\text{-}ABPRE} = \varphi$ (for any $j_\tau \in [1, q_\tau]$), we can build a reduction algorithm $\mathcal{C}$ that breaks the general subgroup decision assumption with advantage φ.*

Proof. $\mathcal{C}$ is going to break the general subgroup decision assumption with sets $Z_0 = \{1, 3\}, Z_1 = \{1, 2, 3\}, Z_2 = \{1\}, Z_3 = \{3\}, Z_4 = \{1, 2\}, Z_5 = \{2, 3\}$ by using $\mathcal{A}$ whom can tell the difference between $Game_{(j-1)_\tau}$ and $Game_{j_\tau}^N$ with advantage φ. $\mathcal{C}$ is given $g_1, g_3, X_1X_2, Y_2Y_3, T$, where T is either a random value of $\mathbb{G}_{p_1p_3}$ or a random value of $\mathbb{G}_{p_1p_2p_3}$. $\mathcal{C}$ generates *param* and *msk* as in the previous proof.

For secret key extraction queries, $\mathcal{C}$ responds the queries as follows. For the first $(j-1)_{sk}$ key queries, $\mathcal{C}$ first calls algorithm *KeyGen* to generate normal key components $K, K', K'', \{K_i\}_{i\in S}$, next chooses $\vartheta \in_R \mathbb{Z}_N$ and sets the semi-functional key as $K(Y_2Y_3)^\vartheta, K', K'', \{K_i\}_{i\in S}$. The key is properly distributed because Y_2^ϑ is a uniformly random value. For the j_{sk}-th key query, $\mathcal{C}$ chooses $\hat{u} \in_R \mathbb{Z}_N$ and $R, R', R'', \{R_i\}_{i\in S} \in_R \mathbb{G}_{p_3}$, and sets the key as $K = g_1^\alpha T^a T^{\hat{u}\kappa} R, K' = T^{\hat{u}} R', K'' = TR'', \forall i \in S\ K_i = T^{h_i} R_i$. $\mathcal{C}$ implicitly sets g^t to be the $\mathbb{G}_{p_1}$ part of T, and g^u to be the $\mathbb{G}_{p_1}$ part of $T^{\hat{u}}$. If T is an element from $\mathbb{G}_{p_1p_2p_3}$, the key above is properly distributed, where $a' = a$ modulo p_2, $\kappa' = \kappa$ modulo p_2, and $\xi_i = h_i$ modulo p_2. For the rest of key queries, $\mathcal{C}$ constructs normal secret keys for $\mathcal{A}$ by calling algorithm *KeyGen* with knowledge of *msk*.

For other types of queries, $\mathcal{C}$ first constructs the corresponding secret keys as in $\mathcal{O}_{sk}$, and next leverages the keys to respond the queries.

In the challenge phase, for the security of original ciphertext, $\mathcal{A}$ outputs $(A^*, \rho^*), m_0, m_1$ to $\mathcal{C}$. $\mathcal{C}$ chooses $\forall j \in [1, l]\ \bar{r}_j \in_R \mathbb{Z}_N$, a vector $\bar{v} = (1, \bar{v}_2, \ldots, \bar{v}_n) \in_R \mathbb{Z}_N^n$, implicitly sets $g^s = X_1$, $v = s\bar{v}$ and $g^{r_j} = X_1^{\bar{r}_j}$. It then sets the ciphertext as

$$B'_0 = m_b \cdot e(g_1, X_1X_2)^\alpha, \quad B'_1 = X_1X_2,$$
$$B'_2 = (X_1X_2)^\kappa, \quad B'_3 = (X_1X_2)^{\hat{\beta}svk+\beta},$$
$$B'_4 = (X_1X_2)^\epsilon, \quad \forall j \in [1, l]$$
$$C'_j = (X_1X_2)^{aA_j^*\bar{v}} (X_1X_2)^{-\bar{r}_j h_{\rho^*(j)}}, \quad D'_j = (X_1X_2)^{\bar{r}_j},$$
$$E' = Sign(ssk, (B'_0, B'_1, B'_3, \forall j \in [1, l]\ (C'_j, D'_j))),$$

where $(ssk, svk) \leftarrow OTS.KeyGen(1^k)$, and we suppose each svk and E' are identically distributed in their corresponding space of *OTS*. This implicitly sets $g_2^{s'} = X_2$ modulo p_2, $\kappa' = \kappa$ modulo p_2, $a' = a$ modulo p_2, $\beta' = \beta$ modulo p_2, $\epsilon' = \epsilon$ modulo p_2, and for each j, $\xi_{\rho^*(j)} = h_{\rho^*(j)}$ modulo p_2, and $g_2^{\gamma_j} = X_2^{\bar{r}_j}$. For the security of re-encrypted ciphertext, $\mathcal{A}$ outputs (A^*, ρ^*), (A, ρ) and m_0, m_1 to $\mathcal{C}$. $\mathcal{C}$ first generates a re-encryption key $rk_{S\to(A^*,\rho^*)}$ by using a normal secret key sk_S and constructs an original ciphertext for m_b under (A, ρ) by calling algorithm *Encrypt*, where $S \models (A, \rho)$. $\mathcal{C}$ then re-encrypts the original ciphertext, and constructs σ_1 as in the real scheme. It then constructs σ_2 as the original ciphertext (excluding B'_4), and outputs the re-encrypted ciphertext (σ_1, σ_2).

If $T \in \mathbb{G}_{p_1p_3}$, this game is in $Game_{(j-1)_\tau}$. If $T \in \mathbb{G}_{p_1p_2p_3}$, then this game is in $Game_{j_\tau}^N$. Thus if $\mathcal{A}$ can tell the difference between $Game_{(j-1)_\tau}$ and $Game_{j_\tau}^N$ with advantage φ, then $\mathcal{C}$ can use $\mathcal{A}$ to break the general subgroup decision assumption with advantage φ. □

Lemma 3. *If there is a PPT algorithm $\mathcal{A}$ such that $Game_{j_\tau}^N Adv_{\mathcal{A}}^{CP\text{-}ABPRE} - Game_{j_\tau}^T Adv_{\mathcal{A}}^{CP\text{-}ABPRE} = \varphi$ for a j_τ belonging to the Phase 1 queries, we can build a reduction algorithm $\mathcal{C}$ that breaks the three party Diffie–Hellman assumption in a subgroup with advantage φ.*

Proof. $\mathcal{C}$ is going to break the three party Diffie–Hellman assumption in a subgroup by using $\mathcal{A}$ whom can notice the difference between $Game_{j_\tau}^N$ and $Game_{j_\tau}^T$ (for the Phase 1 queries) with advantage φ. $\mathcal{C}$ is given $g_1, g_2, g_3, g_2^x, g_2^y, g_2^z, T$, where T is either equal to g_2^{xyz} or a random value of $\mathbb{G}_{p_2}$. $\mathcal{C}$ generates *param* and *msk* as in the proof of Lemma 1.

For secret key extraction queries, $\mathcal{C}$ responds as follows. For the first $(j-1)_{sk}$ key queries, $\mathcal{C}$ first generates the normal key components and next multiplies K by a random element of $\mathbb{G}_{p_2}$. For the j_{sk}-th secret key query, for each $i \in S$ $\mathcal{C}$ chooses $\xi_i \in_R \mathbb{Z}_N$, for $i \notin S$ $\mathcal{C}$ implicitly sets $\xi_i = x\hat{\xi}_i$ modulo p_2, where $\hat{\xi}_i \in_R \mathbb{Z}_N$ for each $i \notin S$. It implicitly sets $a' = xy$ modulo p_2. $\mathcal{C}$ runs algorithm *KeyGen* to generate $K, K', K'', \{K_i\}_{i\in S}$, and next chooses $\kappa', u' \in_R \mathbb{Z}_N$. It implicitly sets $t' = z$ modulo p_2, and sets the key as $Kg_2^{\kappa' u' T}, K' g_2^{u'}, K'' g_2^z, \forall i \in S\ K_i = (g_2^z)^{\xi_i}$. If $T = g_2^{xyz}$, the above key is a properly distributed nominal semi-functional secret key, and otherwise, the key is a properly distributed temporary semi-functional key. For the remaining key queries, $\mathcal{C}$ constructs the normal secret keys by calling algorithm *KeyGen*.

For other types of queries, $\mathcal{C}$ first generates the corresponding secret keys as in $\mathcal{O}_{sk}$, and next leverages the keys to respond the queries.

In the challenge phase, for the security of original ciphertext, given (A^*, ρ^*), m_0, m_1, $\mathcal{C}$ first generates the normal ciphertext components $B_0, B_1, B_2, B_3, B_4,\ \forall j \in [1, l]\ (C_j, D_j)$. Since for each attribute set S issued by $\mathcal{A}$ in the secret key queries satisfying $S \not\models (A^*, \rho^*)$, $\mathcal{C}$ then finds a vector $\bar{v} \in \mathbb{Z}_N^n$ such that $\bar{v}A_j^* = 0$ modulo N, where $\rho^*(j) \in S$ and the first entry of $\bar{v}$ should be nonzero modulo each prime dividing N. We assume that the first entry of $\bar{v}$, denoted as s', is random modulo p_2. $\mathcal{C}$ chooses another vector $v' = (0, v_2', \ldots, v_n') \in_R \mathbb{Z}_N^n$, and implicitly sets the sharing vector $\hat{v}$ to be $a'\hat{v} = xy\bar{v} + v'$, where $\hat{v}$ is randomly distributed as s' is random and the remaining variables of v' are random as well. For each j such that $\rho^*(j) \in S$ $\mathcal{C}$ chooses $\gamma_j \in_R \mathbb{Z}_N$, and for each j such that $\rho^*(j) \notin S$ chooses $\hat{\gamma}_j \in_R \mathbb{Z}_N$. For the j such that $\rho^*(j) \notin S$, $\mathcal{C}$ implicitly sets $\gamma_j = y\hat{\xi}_{\rho^*(j)}^{-1} A_j^* \bar{v} + \hat{\gamma}_j$. It then sets the ciphertext as

$$B_0' = B_0, \qquad B_1' = B_1 g_2^{s'}, \qquad B_2' = B_2 g_2^{\kappa' s'},$$
$$B_3' = B_3 (g_2^{\hat{\beta} svk} g_2^{\beta'})^{s'}, \qquad B_4' = B_4 g_2^{\epsilon' s'}$$
$$\forall j \text{ s.t. } \rho^*(j) \in S, \quad C_j' = C_j g_2^{A_j^* v'} g_2^{-\hat{\xi}_{\rho^*(j)} \gamma_j}, \qquad D_j' = D_j g_2^{\gamma_j},$$
$$\forall j \text{ s.t. } \rho^*(j) \notin S, \quad C_j' = C_j g_2^{A_j^* v'} g_2^{-\hat{\xi}_{\rho^*(j)} \hat{\gamma}_j x}, \qquad D_j' = D_j g_2^{y\hat{\xi}_{\rho^*(j)}^{-1} A_j^* \bar{v} + \hat{\gamma}_j},$$
$$E' = Sign(ssk, (B_0', B_1', B_3',\ \forall j \in [1, l]\ (C_j', D_j'))),$$

where $(ssk, svk) \leftarrow OTS.KeyGen(1^k)$, and we suppose each svk and E' are identically distributed in their corresponding space of OTS. For the security of re-encrypted ciphertext, $\mathcal{A}$ outputs (A^*, ρ^*), (A, ρ) and m_0, m_1 to $\mathcal{C}$. $\mathcal{C}$ first generates a re-encryption key $rk_{S \to (A^*, \rho^*)}$ by using a normal secret key sk_S and constructs an original ciphertext for m_b under (A, ρ) by calling algorithm $Encrypt$, where $S \models (A, \rho)$. $\mathcal{C}$ then re-encrypts the original ciphertext, and constructs σ_1 as in the real scheme. It then constructs σ_2 as the original ciphertext (excluding B_4'), and outputs the re-encrypted ciphertext (σ_1, σ_2).

If $T = g_2^{xyz}$, this game is in $Game_{j_\tau}^N$. If $T \in_R \mathbb{G}_{p_2}$, this game is in $Game_{j_\tau}^T$. Thus if $\mathcal{A}$ can tell the difference between $Game_{j_\tau}^N$ and $Game_{j_\tau}^T$ (for the Phase 1 queries) with advantage φ, $\mathcal{C}$ can use $\mathcal{A}$ to break the three party Diffie–Hellman assumption in a subgroup with advantage φ. □

Lemma 4. *If there is a PPT algorithm $\mathcal{A}$ such that $Game_{j_\tau}^N Adv_{\mathcal{A}}^{CP\text{-}ABPRE} - Game_{j_\tau}^T Adv_{\mathcal{A}}^{CP\text{-}ABPRE} = \varphi$ for a j_τ belonging to the Phase 2 queries, we can build a reduction algorithm $\mathcal{C}$ that breaks the source group q-parallel BDHE assumption in a subgroup with advantage φ.*

Proof. $\mathcal{C}$ is going to break the source group q-parallel BDHE assumption in a subgroup by using $\mathcal{A}$ whom can tell the difference between $Game_{j_\tau}^N$ and $Game_{j_\tau}^T$ (for the Phase 2 queries) with advantage φ. $\mathcal{C}$ is given $g_1, g_2, g_3, g_2^f, g_2^{df},\ \forall i \in [1, 2q] \setminus \{q+1\}\ g_2^{c^i},\ \forall i \in [1, 2q] \setminus \{q+1\}, j \in [1, q]\ g_2^{c^i/b_j},\ \forall j \in [1, q]\ g_2^{dfb_j},\ \forall i \in [1, q], j, j' \in [1, q], j' \neq j,\ g_2^{dfc^i b_{j'}/b_j}, T$, where T is either equal to $g_2^{dc^{q+1}}$ or a random value of $\mathbb{G}_{p_2}$. $\mathcal{C}$ generates $param$ and msk as in the proof of Lemma 1.

Here the challenge ciphertext will be generated before the Phase 2 queries. Thus $\mathcal{C}$ will define the exponents $a', \kappa', \beta', \epsilon', \{\xi_i\}$ after receiving (A^*, ρ^*) from $\mathcal{A}$. $\mathcal{C}$ chooses $\hat{\kappa}, \{\hat{\xi}_i\}_{i \in \mathcal{U}} \in_R \mathbb{Z}_N$, and implicitly sets $a' = cd$ modulo p_2 and $\kappa' = d + \hat{k}$ modulo p_2. $\mathcal{C}$ sets $g_2^{\xi_i}$ to be $g_2^{\hat{\xi}_i} \prod_{j \in J_i} g_2^{A_{j,1}^* c/b_j} \cdots g_2^{A_{j,n}^* c^n/b_j}$, where J_i is the set of indices j such that $\rho^*(j) = i$. For each $j \in [1, l]$ $\mathcal{C}$ chooses $\hat{\gamma}_j \in_R \mathbb{Z}_N$, and generates the normal ciphertext components $B_0, B_1, B_2, B_3, B_4,\ \forall j \in [1, l]\ (C_j, D_j)$ as in the algorithm $Encrypt$. It implicitly sets $s' = f$ modulo p_2, and $\gamma_j = dfb_j + \hat{\gamma}_j$. It further chooses $y_2, \ldots, y_n \in_R \mathbb{Z}_N$ and implicitly sets a vector $\hat{v} = (f, fc + y_2 a'^{-1}, \ldots, fc^{n-1} + y_n a'^{-1})$. For the security of original ciphertext, $\mathcal{C}$ sets the ciphertext as

$$B_0' = B_0, \qquad B_1' = B_1 g_2^f, \qquad B_2' = B_2 g_2^{f(d+\hat{\kappa})},$$
$$B_3' = B_3 g_2^{f\hat{\beta} svk} g_2^{f\beta'},$$
$$B_4' = B_4 g_2^{f\epsilon'} \quad \forall j \in [1, l] \qquad C_j' = C_j \Gamma, \qquad D_j' = D_j g_2^{dfb_j + \hat{\gamma}_j},$$
$$E' = Sign(ssk, (B_0', B_1', B_3',\ \forall j \in [1, l]\ (C_j', D_j'))),$$

where set $\Gamma = g_2^{y_2 A_{j,2}^* + \cdots + y_n A_{j,n}^*} \cdot g_2^{-\sum_{j' \in J_{\rho^*(j)},\ j \neq j'} (dfcA_{j',1}^* b_j/b_{j'} + \cdots + dfc^n A_{j',n}^* b_j/b_{j'})} \cdot g_2^{-dfb_j \hat{\xi}_{\rho^*(j)} - \hat{\gamma}_j \hat{\xi}_{\rho^*(j)}} \cdot g_2^{-\hat{\gamma}_j (\sum_{j' \in J_{\rho^*(j)}} cA_{j',1}^*/b_{j'} + \cdots + c^n A_{j',n}^*/b_{j'})}$, $\beta', \epsilon' \in_R \mathbb{Z}_N$, $(ssk, svk) \leftarrow OTS.KeyGen(1^k)$, and we suppose each svk and E' are identically distributed in their corresponding space of OTS. For the security of re-encrypted ciphertext, $\mathcal{A}$ outputs (A^*, ρ^*), (A, ρ) and m_0, m_1 to $\mathcal{C}$. $\mathcal{C}$ first generates a re-encryption key $rk_{S \to (A^*, \rho^*)}$ by using a normal secret key sk_S and constructs an original ciphertext for m_b under (A, ρ) by calling algorithm $Encrypt$, where $S \models (A, \rho)$. $\mathcal{C}$ then re-encrypts the original ciphertext, and constructs σ_1 as in the real scheme. It then constructs σ_2 as the original ciphertext (excluding B_4'), and outputs the re-encrypted ciphertext (σ_1, σ_2).

For secret key extraction queries, $\mathcal{C}$ responds as follows. For the first $(j-1)_{sk}$ key queries, $\mathcal{C}$ first generates the normal key components and next multiplies K by a random element of $\mathbb{G}_{p_2}$. For the j_{sk}-th secret key query, $\mathcal{C}$ calls the algorithm $KeyGen$ to generate normal key components $(K, K', K'',\ \forall i \in S\ K_i)$. It then chooses a vector $\phi = (\phi_1, \ldots, \phi_n) \in_R \mathbb{Z}_N^n$ such that $\phi A_j^* = 0$ modulo N, where $\rho^*(j) \in S$ and the first entry of ϕ should be nonzero modulo each prime dividing N. $\mathcal{C}$ chooses $\hat{u} \in_R \mathbb{Z}_N$ and sets $K' g_2^{u'} = K' g_2^{-\phi_2 c^q - \phi_3 c^{q-1} - \cdots - \phi_n c^{q-n+2} + f\hat{u}}$, $K'' g_2^{t'} = K'' g_2^{\phi_1 c^q + \phi_2 c^{q-1} + \cdots + \phi_n c^{q-n+1}}$. Since ϕ is orthogonal to A_j^* $(\rho(j)^* \in S)$, $\mathcal{C}$ then sets $K_i g_2^{t'\xi_i} = K_i g_2^{t'\hat{\xi}_i + \sum_{j \in J_i} \sum_{\zeta_1, \zeta_2 = 1, \zeta_1 \neq \zeta_2}^{n} \phi_{\zeta_1} A_{j,\zeta_2}^* b_j^{-1} c^{q+1+\zeta_2-\zeta_1}}$, where $q + 1 + \zeta_2 - \zeta_1 \in [1, 2q] \setminus \{q+1\}$. Finally, $\mathcal{C}$ sets $K = T^{\phi_1} g_2^{-\hat{\kappa} c^q \phi_2 \cdots - \hat{\kappa} c^{q-n+2} \phi_n + df\hat{u} + f\hat{\kappa}\hat{u}}$. For the remaining secret key queries, $\mathcal{C}$ constructs the normal keys by calling algorithm $KeyGen$.

For other types of queries, $\mathcal{C}$ first generates the corresponding secret keys as above, and next leverages the keys to respond the queries. If $T = g_2^{dc^{q+1}}$, this game is in $Game_{j_\tau}^N$. If $T \in_R \mathbb{G}_{p_2}$, this game is in $Game_{j_\tau}^T$. Thus if $\mathcal{A}$ can tell the difference between $Game_{j_\tau}^N$ and $Game_{j_\tau}^T$ (for the Phase 2 queries) with advantage φ, $\mathcal{C}$ can use $\mathcal{A}$ to break the source group q-parallel BDHE assumption in a subgroup with advantage φ. □

Lemma 5. *If there is a PPT algorithm $\mathcal{A}$ such that $Game_{j_\tau}^T Adv_{\mathcal{A}}^{CP\text{-}ABPRE} - Game_{j_\tau} Adv_{\mathcal{A}}^{CP\text{-}ABPRE} = \varphi$ (for any $j_\tau \in [1, q_\tau]$), we can build a reduction algorithm $\mathcal{C}$ that breaks the general subgroup decision assumption with advantage φ.*

Proof. This proof is almost identical to the proof of Lemma 2 but with the exception that $\mathcal{C}$ will use $Y_2 Y_3$ to randomize the $\mathbb{G}_{p_2}$ part of K such that the key is not nominal semi-functional. □

Lemma 6. *If there is a PPT algorithm $\mathcal{A}$ such that $Game_q Adv_{\mathcal{A}}^{CP\text{-}ABPRE} - Game_{final} Adv_{\mathcal{A}}^{CP\text{-}ABPRE} = \varphi$, we can build a reduction algorithm $\mathcal{C}$ that breaks Assumption 1 with advantage φ.*

Proof. $\mathcal{C}$ is going to break the Assumption 1 by using $\mathcal{A}$ whom can tell the difference between $Game_q$ and $Game_{final}$ with advantage φ. $\mathcal{C}$ is given $g_1, g_2, g_3, g_1^\alpha X_2, g_1^s Y_2, T$, where T is either equal to $(g_1, g_1)^{\alpha s}$ or a random value of $\mathbb{G}_T$. $\mathcal{C}$ chooses $a, \kappa, \beta, \hat{\beta}, \epsilon, \{h_i\}_{i \in \mathcal{U}} \in_R \mathbb{Z}_N$ and sets the $param$ as $(N, g = g_1, \hat{g}_1$

$= g_1^{\hat{\beta}}, g^a = g_1^a, g^\kappa = g_1^\kappa, g^\beta = g_1^\beta, g^\epsilon = g_1^\epsilon, e(g,g)^\alpha = e(g_1^\alpha X_2, g_1),\ \forall i \in \mathcal{U}\ H_i = g^{h_i} = g_1^{h_i}, TCR_1, TCR_2, SYM, OTS)$, where $\mathcal{C}$ chooses TCR_1, TCR_2, SYM, OTS as in the real scheme. Here the master secret key is unknown to $\mathcal{C}$.

For secret key extraction queries, $\mathcal{C}$ responds the semi-functional keys to $\mathcal{A}$ as follows. It chooses $t, u, \gamma \in_R \mathbb{Z}_N, R, R', R'', \{R_i\}_{i\in\mathcal{U}} \in_R \mathbb{G}_{p_3}$, and sets the key as $K = g_1^\alpha X_2 g_1^{at} g_1^{\kappa u} R g_2^\gamma, K' = g_1^u R', K'' = g_1^t R'',\ \forall i \in S\ K_i = g_1^{h_i t}$. For other types of queries, $\mathcal{C}$ first generates the corresponding semi-functional secret keys, and next leverages the keys to respond the queries.

In the challenge phase, for the security of original ciphertext, $\mathcal{A}$ outputs $(A^*, \rho^*), m_0, m_1$ to $\mathcal{C}$. $\mathcal{C}$ chooses $\forall j \in [1, l]\ \bar{r}_j \in_R \mathbb{Z}_N$, a vector $\bar{v} = (1, \bar{v}_2, \ldots, \bar{v}_n) \in_R \mathbb{Z}_N^n$, implicitly sets $v = s\bar{v}$ and $r_j = s\bar{r}_j$. It then sets the ciphertext as

$$B'_0 = m_b T, \qquad B'_1 = g_1^s Y_2, \qquad B'_2 = (g_1^s Y_2)^\kappa,$$

$$B'_3 = (g_1^s Y_2)^{\hat{\beta} svk + \beta}, \qquad B'_4 = (g_1^s Y_2)^\epsilon,$$

$$\forall j \in [1, l] \quad C'_j = (g_1^s Y_2)^{a A_j^* \bar{v}} (g_1^s Y_2)^{-\bar{r}_j h_{\rho^*(j)}}, \qquad D'_j = (g_1^s Y_2)^{\bar{r}_j},$$

$$E' = Sign(ssk, (B'_0, B'_1, B'_3,\ \forall j \in [1, l]\ (C'_j, D'_j))),$$

where $(ssk, svk) \leftarrow OTS.KeyGen(1^k)$, and we suppose each svk and E' are identically distributed in their corresponding space of OTS. This implicitly sets $g_2^{s'} = Y_2$ modulo p_2, $\kappa' = \kappa$ modulo p_2, $a' = a$ modulo p_2, $\hat{\beta} = \beta''$ modulo p_2, $\beta' = \beta$ modulo p_2, $\epsilon' = \epsilon$ modulo p_2, $\hat{v} = s'\bar{v}$ modulo p_2, and for each j, $\xi_{\rho^*(j)} = h_{\rho^*(j)}$ modulo p_2, and $g_2^{\gamma_j} = Y_2^{\bar{r}_j}$ modulo p_2. For the security of re-encrypted ciphertext, $\mathcal{A}$ outputs (A^*, ρ^*), (A, ρ) and m_0, m_1 to $\mathcal{C}$. $\mathcal{C}$ first generates a re-encryption key $rk_{S\to(A^*,\rho^*)}$ by using a semi-functional secret key sk_S and constructs an original (normal) ciphertext for m_b under (A, ρ) by calling algorithm $Encrypt$, where $S \models (A, \rho)$. $\mathcal{C}$ then re-encrypts the original (normal) ciphertext, and constructs σ_1 as in the real scheme. It then constructs σ_2 as the above original (semi-functional) ciphertext (excluding B'_4), and outputs the re-encrypted ciphertext (σ_1, σ_2).

If $T = e(g_1, g_1)^{\alpha s}$, this game is in $Game_q$. If $T \in_R \mathbb{G}_T$, then this game is in $Game_{final}$. Thus if $\mathcal{A}$ can tell the difference between $Game_q$ and $Game_{final}$ with advantage φ, then $\mathcal{C}$ can use $\mathcal{A}$ to break the Assumption 1 with advantage φ. □

This completes the proof of Theorem 1.

6. Applications of CP-ABPRE

6.1. Secure cloud storage data sharing

Our system is applicable to many practical cloud-based settings. Suppose a cloud storage system user (acting as a data manager), say Alice, will store some project data to the cloud server in an encrypted format so as to share the data with her teammates. For example, she encrypts the data to the cloud under an access policy P_1 = ("*Department* : *Science Research*" *and* "*Position* : *Team worker or above*" *and* "*Project* : *A*" *and* "*Team* : 001"), that is, any of her teammate/team leader (in Team 001) under Project A in the Science Research Department is allowed to access the data. Nevertheless, to improve the processing speed of the project, the company (Alice working at) decides to request another working team 002 in Software Develop Department to join into the project. Accordingly, the project data needs to be shared with the team 002. Here Alice requests the PKG to generate a re-encryption key for a new access policy P_2 = ("*Department* : *Science Research*, *Software Develop*" *and* "*Position* : *Team worker or above*" *and* "*Project* : *A*" *and* "*Team* : 001, 002"). The cloud storage system server then uses the re-encryption key to convert the ciphertexts under P_1 to new ciphertexts under P_2 such that the original (Alice) team but also the new joining team can access the project data.

6.2. Cloud computing data secure forwarding

In many cloud computing scenarios, data which to be computed is usually distributed stored in different places. Some sensitive data might be encrypted before its delivery. For example, before sending a group of personal biometric data, a data sender has rights to choose to which cloud computing center the data will be sent. Suppose the sender prefers to allow computing centers satisfying the following properties to handle the data: CP_1 = ("*Speed of Calculation* : *nanoseconds*" *and* "*Location* : *domestic*" *and* "*Security Level* : *Confidential*" *and* "*Support* : *Educational Institute or Industry*"). The sender then encrypts the biometric data under CP_1, and next uploads the encryption to the cloud such that any computing center matching CP_1 can read and compute the data. Suppose there is a university computing center, say UCC, satisfying CP_1 but it is out of service. Accordingly, the task of data computing might be suspended. Here CP-ABPRE comes to help. Before it is unavailable, UCC first generates a re-encryption key from its attribute sets to a new access policy CP_2 = ("*Speed of Calculation* : *nanoseconds or above*" *and* "*Location* : *No Preference*" *and* "*Security Level* : *Confidential or above*" *and* "*Support* : *Educational Institute*"), and then uploads the key to its proxy. When UCC is either too busy to response a computing task or in the maintenance period, the proxy will directly re-encrypt the encryption of the data under CP_1 (sent to the center) to another encryption under CP_2 such that some available computing centers matching CP_2 can proceed the computing task on behalf of UCC.

6.3. Social network profile secure sharing

Our system can be also deployed into social network settings. For instance, a social network user (such as LinkedIn, Facebook), say Alice, might choose to share her profile with others in a specified way. She may encrypt her profile under an access policy PP_1 = ("*Age* : *No Preference*" *and* "*Location* : *Local*" *and* "*Gender* : *Female*" *and* "*Status* : *Student*"), and then upload the encrypted profile to the social network. If the network users with attributes matching PP_1, they can read Alice's profile. When Alice updates her profile access policy, she only needs to generate (or request the PKG to generate when using a resource-limited device) a re-encryption key for a new policy, e.g., PP_2 = ("*Age* : *No Preference*" *and* "*Location* : *Local and Oversea*" *and* "*Gender* : *No Preference*" *and* "*Status* : *HR and Student*"). Given the re-encryption key, the social network server will automatically update the old version of encrypted profile to the new one such that other social network users can read the profile if they matches PP_2.

7. Conclusions

In this paper we defined the IND-CCA security notion for CP-ABPRE systems, and proposed the first adaptively CCA-secure CP-ABPRE scheme without loss of expressiveness on access policy by integrating the dual system encryption technology with selective proof technique. Following the proof framework introduced by Lewko and Waters, we proved our scheme in the standard model.

This paper also motivates some interesting open problems, for example, how to convert our system in the prime order bilinear group.

Acknowledgments

W. Susilo is partially supported by the Australian Research Council Linkage Project LP120200052. D.S. Wong is supported by a grant from the RGC of the HKSAR, China (Project No. CityU 121512). Y. Yu is supported by the Vice Chancellor's research fellowship of University of Wollongong and the NSFC of China under Grant 61003232.

References

[1] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: A. Juels, R.N. Wright, S.D.C. di Vimercati (Eds.), ACM Conference on Computer and Communications Security, ACM, 2006, pp. 89–98.
[2] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: R. Cramer (Ed.), Advances in Cryptology—EUROCRYPT 2005, in: LNCS, vol. 3494, Springer, Berlin, Heidelberg, 2005, pp. 457–473. http://dx.doi.org/10.1007/11426639_27.
[3] M. Mambo, E. Okamoto, Proxy cryptosystems: delegation of the power to decrypt ciphertexts, IEICE Trans. E80-A (1) (1997) 54–63.
[4] M. Blaze, G. Bleumer, M. Strauss, Divertible protocols and atomic proxy cryptography, in: K. Nyberg (Ed.), EUROCRYPT, in: LNCS, vol. 1403, Springer, 1998, pp. 127–144.
[5] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, ACM Trans. Inf. Syst. Secur. 9 (1) (2006) 1–30.
[6] X. Liang, Z. Cao, H. Lin, J. Shao, Attribute based proxy re-encryption with delegating capabilities, in: W. Li, W. Susilo, U.K. Tupakula, R. Safavi-Naini, V. Varadharajan (Eds.), ASIACCS, ACM, 2009, pp. 276–286.
[7] K. Liang, L. Fang, W. Susilo, D.S. Wong, A ciphertext-policy attribute-based proxy re-encryption with chosen-ciphertext security, in: INCoS, IEEE, 2013, pp. 552–559.
[8] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: IEEE Symposium on Security and Privacy, IEEE Computer Society, 2007, pp. 321–334.
[9] L. Cheung, C.C. Newport, Provably secure ciphertext policy ABE, in: P. Ning, S.D.C. di Vimercati, P.F. Syverson (Eds.), ACM Conference on Computer and Communications Security, ACM, 2007, pp. 456–465.
[10] V. Goyal, A. Jain, O. Pandey, A. Sahai, Bounded ciphertext policy attribute based encryption, in: L. Aceto, I. Damgård, L.A. Goldberg, M.M. Halldórsson, A. Ingólfsdóttir, I. Walukiewicz (Eds.), ICALP (2), in: LNCS, vol. 5126, Springer, 2008, pp. 579–591.
[11] B. Waters, Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization, in: D. Catalano, N. Fazio, R. Gennaro, A. Nicolosi (Eds.), Public Key Cryptography, in: LNCS, vol. 6571, Springer, 2011, pp. 53–70.
[12] N. Attrapadung, J. Herranz, F. Laguillaumie, B. Libert, E. de Panafieu, C. Rafols, Attribute-based encryption schemes with constant-size ciphertexts, Theoret. Comput. Sci. 422 (2012) 15–38.
[13] B. Waters, Functional encryption for regular languages, in: R. Safavi-Naini, R. Canetti (Eds.), CRYPTO, in: LNCS, vol. 7417, Springer, 2012, pp. 218–235.
[14] J. Li, K. Kim, Hidden attribute-based signatures without anonymity revocation, Inform. Sci. 180 (9) (2010) 1681–1689. http://dx.doi.org/10.1016/j.ins.2010.01.008.
[15] J. Li, Q. Wang, C. Wang, K. Ren, Enhancing attribute-based encryption with attribute hierarchy, MONET 16 (5) (2011) 553–561. http://dx.doi.org/10.1007/s11036-010-0233-y.
[16] J. Li, X. Huang, J. Li, X. Chen, Y. Xiang, Securely outsourcing attribute-based encryption with checkability, IEEE Trans. Parallel Distrib. Syst. 25 (8) (2014) 2201–2210. http://dx.doi.org/10.1109/TPDS.2013.271.
URL: http://doi.ieeecomputersociety.org/10.1109/TPDS.2013.271.
[17] A.B. Lewko, T. Okamoto, A. Sahai, K. Takashima, B. Waters, Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption, in: H. Gilbert (Ed.), EUROCRYPT, in: LNCS, vol. 6110, Springer, 2010, pp. 62–91.
[18] A.B. Lewko, B. Waters, New proof methods for attribute-based encryption: achieving full security through selective techniques, in: R. Safavi-Naini, R. Canetti (Eds.), CRYPTO, in: LNCS, vol. 7417, Springer, 2012, pp. 180–198.
[19] R. Canetti, S. Hohenberger, Chosen-ciphertext secure proxy re-encryption, in: P. Ning, S.D.C. di Vimercati, P.F. Syverson (Eds.), ACM Conference on Computer and Communications Security, ACM, 2007, pp. 185–194.
[20] T. Isshiki, M.H. Nguyen, K. Tanaka, Proxy re-encryption in a stronger security model extended from CT-RSA2012, in: CT-RSA 2012, in: LNCS, vol. 7779, Springer, 2013, pp. 277–292.
[21] B. Libert, D. Vergnaud, Unidirectional chosen-ciphertext secure proxy re-encryption, in: R. Cramer (Ed.), Public Key Cryptography, in: LNCS, vol. 4939, Springer, 2008, pp. 360–379.
[22] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang, Y. Zhao, Generic construction of chosen ciphertext secure proxy re-encryption, in: O. Dunkelman (Ed.), Topics in Cryptology—CT-RSA 2012, in: LNCS, vol. 7178, Springer, Berlin, Heidelberg, 2012, pp. 349–364. http://dx.doi.org/10.1007/978-3-642-27954-6_22.
[23] J. Weng, M. Chen, Y. Yang, R.H. Deng, K. Chen, F. Bao, CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles, Sci. China Inf. Sci. 53 (3) (2010) 593–606.
[24] K. Liang, C. Chu, X. Tan, D.S. Wong, C. Tang, J. Zhou, Chosen-ciphertext secure multi-hop identity-based conditional proxy re-encryption with constant-size ciphertexts, Theoret. Comput. Sci. 539 (2014) 87–105. http://dx.doi.org/10.1016/j.tcs.2014.04.027.
[25] K. Liang, J.K. Liu, D.S. Wong, W. Susilo, An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing, in: M. Kutylowski, J. Vaidya (Eds.), Computer Security—ESORICS 2014—19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7–11, 2014. Proceedings, Part I, in: Lecture Notes in Computer Science, vol. 8712, Springer, 2014, pp. 257–272. http://dx.doi.org/10.1007/978-3-319-11203-9_15.
[26] K. Liang, Z. Liu, X. Tan, D.S. Wong, C. Tang, A CCA-secure identity-based conditional proxy re-encryption without random oracles, in: T. Kwon, M.-K. Lee, D. Kwon (Eds.), ICISC, in: LNCS, vol. 7839, Springer, 2012, pp. 231–246.
[27] K. Liang, Q. Huang, R. Schlegel, D.S. Wong, C. Tang, A conditional proxy broadcast re-encryption scheme supporting timed-release, in: R.H. Deng, T. Feng (Eds.), ISPEC, in: Lecture Notes in Computer Science, vol. 7863, Springer, 2013, pp. 132–146.
[28] R. Lu, X. Lin, J. Shao, K. Liang, Rcca-secure multi-use bidirectional proxy re-encryption with master secret security, in: S.S.M. Chow, J.K. Liu, L.C.K. Hui, S. Yiu (Eds.), Provable Security—8th International Conference, ProvSec 2014, Hong Kong, China, October 9–10, 2014. Proceedings, in: Lecture Notes in Computer Science, vol. 8782, Springer, 2014, pp. 194–205. http://dx.doi.org/10.1007/978-3-319-12475-9_14.
[29] S. Luo, J. Hu, Z. Chen, Ciphertext policy attribute-based proxy re-encryption, in: M. Soriano, S. Qing, J. López (Eds.), ICICS, in: LNCS, vol. 6476, Springer, 2010, pp. 401–415.
[30] T. Mizuno, H. Doi, Hybrid proxy re-encryption scheme for attribute-based encryption, in: F. Bao, M. Yung, D. Lin, J. Jing (Eds.), Information Security and Cryptology, in: LNCS, vol. 6151, Springer, Berlin, Heidelberg, 2011, pp. 288–302. http://dx.doi.org/10.1007/978-3-642-16342-5_21.
[31] N. Chandran, M. Chase, V. Vaikuntanathan, Functional re-encryption and collusion-resistant obfuscation, in: R. Cramer (Ed.), TCC, in: Lecture Notes in Computer Science, vol. 7194, Springer, 2012, pp. 404–421.
[32] K. Liang, M.H. Au, W. Susilo, D.S. Wong, G. Yang, Y. Yu, An adaptively CCA-secure ciphertext-policy attribute-based proxy re-encryption for cloud data sharing, in: X. Huang, J. Zhou (Eds.), ISPEC, in: Lecture Notes in Computer Science, vol. 8434, Springer, 2014, pp. 448–461.
[33] A. Beimel, Secure schemes for secret sharing and key distribution (Ph.D. thesis), Israel Institute of Technology, Technion, Haifa, Israel, 1996.
[34] D. Boneh, E.-J. Goh, K. Nissim, Evaluating 2-DNF formulas on ciphertexts, in: J. Kilian (Ed.), TCC, in: LNCS, vol. 3378, Springer, 2005, pp. 325–341.
[35] R. Cramer, V. Shoup, Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack, SIAM J. Comput. 33 (1) (2004) 167–226.
[36] M. Bellare, S. Shoup, Two-tier signatures, strongly unforgeable signatures, and Fiat–Shamir without random oracles, in: T. Okamoto, X. Wang (Eds.), Public Key Cryptography, in: LNCS, vol. 4450, Springer, 2007, pp. 201–216.
[37] R. Canetti, S. Halevi, J. Katz, Chosen-ciphertext security from identity-based encryption, in: C. Cachin, J. Camenisch (Eds.), EUROCRYPT, in: LNCS, vol. 3027, Springer, 2004, pp. 207–222.
[38] R. Canetti, H. Krawczyk, J.B. Nielsen, Relaxing chosen-ciphertext security, in: D. Boneh (Ed.), CRYPTO, in: LNCS, vol. 2729, Springer, 2003, pp. 565–582.
[39] C.-K. Chu, W.-G. Tzeng, Identity-based proxy re-encryption without random oracles, in: J.A. Garay, A.K. Lenstra, M. Mambo, R. Peralta (Eds.), ISC, in: LNCS, vol. 4779, Springer, 2007, pp. 189–202.
[40] K. Emura, A. Miyaji, K. Omote, A timed-release proxy re-encryption scheme, IEICE Trans. 94-A (8) (2011) 1682–1695.

Kaitai Liang received the B.Eng. degree in Software Engineering, the M.S. degree in Computer Applied Technology from South China Agricultural University, China. He is currently a Ph.D. candidate in the Department of Computer Science, City University of Hong Kong. His research interest is applied cryptography; in particular, cryptographic protocols, encryption/signature, and RFID. He is also interested in cybersecurity, such as network security, database security, and security in cloud computing.

Man Ho Au obtained his bachelors and masters degrees from the Department of Information Engineering, Chinese University of Hong Kong. He received his Ph.D. from the University of Wollongong (UOW). Currently, he is an assistant professor at the Department of Computing, Hong Kong Polytechnic University (PolyU). Before joining PolyU, he was a lecturer in UOW. He works in the area of information security. In particular, his research interests include applying public-key cryptographic techniques to systems with security and privacy concerns.

Joseph K. Liu received the Ph.D. degree in Information Engineering from the Chinese University of Hong Kong in July 2004, specializing in cryptographic protocols for securing wireless networks, privacy, authentication and provable security. He is now a research scientist in the Infocomm Security Department at the Institute for Infocomm Research, Singapore. His current technical focus is particularly cybersecurity in physical systems including cloud computing environment, transportation system and smart city infrastructure; lightweight security and privacy enhanced technology.

Willy Susilo received the Ph.D. degree in Computer Science from the University of Wollongong, Wollongong, Australia. He is a Professor with the School of Computer Science and Software Engineering and the Director of Centre for Computer and Information Security Research, University of Wollongong. His main research interests include cryptography and information security.

Duncan S. Wong received the B.Eng. degree from the University of Hong Kong in 1994, the M.Phil. degree from the Chinese University of Hong Kong in 1998, and the Ph.D. degree from Northeastern University, Boston, MA, in 2002. He is currently an associate professor in the Department of Computer Science at the City University of Hong Kong. His primary research interest is cryptography; in particular, cryptographic protocols, encryption and signature schemes, and anonymous systems. He is also interested in other topics in information security, such as network security, wireless security database security, and security in cloud computing.

Guomin Yang received his Ph.D. from the Department of Computer Science, City University of Hong Kong in 2009. From 2009 to 2012, he worked as a Research Scientist at the Temasek Laboratories, National University of Singapore. He is currently a Lecture at the School of Computer Science and Software Engineering, University of Wollongong.

Yong Yu received the Ph.D. degree in Cryptography from Xidian University in 2008. He is currently an associate professor of University of Electronic Science and Technology of China. His research interests are cryptography and its applications, especially public encryption, digital signature and secure cloud storage.

Anjia Yang received the B.Sc. degree from Jilin University in 2011. He is currently working toward the Ph.D. degree in the Department of Computer Science at the City University of Hong Kong. His research interests include RFID security and privacy, Cloud Security, and applied cryptography.