Theoretical Computer Science 539 (2014) 87–105

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Chosen-ciphertext secure multi-hop identity-based conditional proxy re-encryption with constant-size ciphertexts

Kaitai Liang [a,*], Cheng-Kang Chu [b], Xiao Tan [a], Duncan S. Wong [a,*], Chunming Tang [c], Jianying Zhou [b]

[a] Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong Special Administrative Region
[b] Infocomm Security Department, Institute for Infocomm Research, 1 Fusionopolis Way, Singapore 138632, Singapore
[c] School of Mathematics and Information Science, Guangzhou University, 230 Wai Huan Xi Road, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 13 March 2013
Received in revised form 15 February 2014
Accepted 22 April 2014
Communicated by X. Deng

Keywords:
Conditional proxy re-encryption
Identity-based proxy re-encryption
Bilinear map
Chosen-ciphertext security
Constant size ciphertext

ABSTRACT

Proxy Re-Encryption (PRE) allows one user to delegate the decryption rights of his/her ciphertexts to another user. Since the introduction of Multi-Hop Identity-Based PRE (MH-IBPRE) by Green and Ateniese, the ciphertext size and the decryption complexity *grow linearly* in the number of re-encryption "hops". In this paper, for the first time, we propose an MH-IBPRE that maintains the (constant) ciphertext size and computational complexity regardless of the number of re-encryption hops. Moreover, our scheme is bidirectional and also supports conditional re-encryption. The scheme is proven secure against selective identity and chosen-ciphertext attacks and collusion resistant in the standard model. As of independent interest, we also show that the conditional re-encryption can also be extended to a set of conditions.

1. Introduction

Public Key Encryption (PKE) is a useful cryptographic primitive, whereby it allows a user to encrypt data under the public key of a receiver such that only the legitimate receiver with the corresponding private key can access the data. In social networks, a data is often shared among different users. A user, say Alice, can share a data, e.g., a picture or video with her friend, say Bob, without loss of confidentiality by using the traditional PKE. Sometimes, Bob might choose to further share the same data with another user, say Carol. In the context of PKE, Bob should first decrypt the ciphertext of the data sent by Alice, and next re-encrypt the data to Carol so as to finish data sharing. This, nevertheless, does not scale well when Bob is off-line or unavailable. An alternative way is that Bob delegates a proxy to encrypt the data to Carol when he is absent. However, this kind of delegation relies on either the accessibility of the data or knowledge of Bob's private key in the view of proxy.

To increase the flexibility of data sharing in the context of PKE, Blaze, Bleumer and Strauss [3] defined the notion of Proxy Re-Encryption (PRE), in which a *semi-trusted proxy* is allowed to transform a ciphertext intended for Alice into another ciphertext of the same plaintext intended for Bob using a given re-encryption key without accessing the underlying plaintext. If the re-encryption key allows the proxy to transform ciphertexts intended for Alice (i.e. *delegator*) to ciphertexts intended for Bob (i.e. *delegatee*) and vice versa, the scheme is *bidirectional*. Whereas, if the re-encryption key only supports

* Corresponding author. Tel.: +852 3442 8020/3442 9719; fax: +852 3442 0503.
E-mail addresses: kliang4-c@my.cityu.edu.hk (K. Liang), chengkangchu@gmail.com (C.-K. Chu), xiaotan4@gapps.cityu.edu.hk (X. Tan), duncan@cityu.edu.hk (D.S. Wong), ctang@gzhu.edu.cn (C. Tang), jyzhou@i2r.a-star.edu.sg (J. Zhou).

http://dx.doi.org/10.1016/j.tcs.2014.04.027
0304-3975/

the transformation from Alice to Bob (resp. from Bob to Alice), the scheme is *unidirectional*. PRE further comes to two flavors: one is *single-hop* PRE, and the other is *multiple-hop* PRE. If a ciphertext can be re-encrypted from Alice to Bob and cannot be further converted, the scheme is single hop. In multi-hop setting, a ciphertext can be re-encrypted from Alice to Bob and to Carol, and so on. The latter might be more desirable than the former in practice as it provides the flexibility of re-delegation, that is, the delegatee can re-delegate the ciphertexts to another users. PRE is applicable to many network applications, such as secure distributed files systems [1] and cloud storage systems (such as SugarSync[1] and Box[2]).

To implement PRE in the identity-based cryptographic setting with multi-hop property, Green and Ateniese [14] defined the notion of multi-hop identity-based proxy re-encryption (MH-IBPRE), and proposed a concrete scheme satisfying the new notion. The scheme allows a proxy to re-encrypt ciphertexts under an identity, e.g., ID_{Alice}, to another identity, e.g., ID_{Bob}, such that Bob can also decrypt, while the proxy can further re-encrypt the ciphertexts intended for a new identity, say ID_{Carol}, and so on. Since the introduction by Green and Ateniese [14], there are a few MH-IBPRE schemes that have been proposed in the literature.

Motivation. MH-IBPRE explores the applications of PRE in practice. In recent years, many Internet users and companies choose to store their data in cloud storage systems due to its considerable storage space. We here use cloud storage systems as an example to illustrate the application for MH-IBPRE so as to motivate our work. Suppose a group A of N employees will share some of their data mutually to fulfill a business project cooperatively. By employing MH-IBPRE, the data sharing can be fulfilled efficiently as follows. Without loss of confidentiality, each employee (e.g., $A.1$) might first encrypt his/her data (e.g., $m_{A.1}$) under the identity (e.g., $ID_{A.1}$) before uploading to the cloud. To share $m_{A.1}$ with another employee, say $A.2$, $A.1$ may generate a re-encryption key $rk_{A.1\rightarrow A.2}$ (from $A.1$ to $A.2$) for the cloud server (acting as a proxy) such that the server can further re-encrypt the ciphertexts of $A.1$ to $A.2$ (e.g., $Enc(ID_{A.1}, m_{A.1}) \rightarrow Enc(ID_{A.2}, m_{A.1})$). When $m_{A.1}$ is further shared with others (e.g., $A.3$), $A.2$ will upload a new re-encryption key $rk_{A.2\rightarrow A.3}$ to the cloud. The cloud then performs the corresponding conversion for $A.3$.

Although MH-IBPRE is proposed to employ the delegation of decryption rights in the context of IBE without losing confidentiality, it yields a price that both the size of ciphertext and decryption complexity grow linearly in the number of re-encryption hops. For example, if an original ciphertext $Enc(ID_{A.1}, m_{A.1})$ is chosen to be delegated via the direction $A.1 \rightarrow A.2 \rightarrow A.3 \rightarrow A.4$, then the size of the re-encrypted ciphertext (as well as decryption complexity) for $A.4$ will be triple larger than that of the original ciphertext. This is undesirable in practice because of the incurred linear communication bandwidth, storage cost and computation complexity.

Like traditional PRE, MH-IBPRE also incurs a potential risk for access control as the re-encryption power of a proxy cannot be controlled precisely. Generally speaking, given a re-encryption key $rk_{A.1\rightarrow A.2}$ the proxy is allowed to re-encrypt *all* $A.1$'s ciphertexts stored in the cloud to $A.2$ without any discrimination. This might contradict $A.1$'s will because $A.1$ might only prefer to share some data tagged with a specified condition, e.g., "public", other than the data labeled with "private" with $A.2$. Furthermore the sharing data might be described by a set of conditions other than a single one. For instance, $A.1$ shares the data, which is uploaded to the cloud in "July" containing keywords "meeting, project", with $A.2$. Here the condition set associated with the sharing data is seen as $W = \{July, meeting, project\}$.

As some data is allowed to be shared mutually among different employees, it indicates that the re-encryption for the ciphertext of the data should be considered in a bidirectional way (e.g., $A.1 \leftrightarrow A.2$). It will bring convenience for the data sharing if a pair of delegator and delegatee uses a bidirectional re-encryption key instead of two separated unidirectional ones. For example, given a re-encryption key $rk_{A.1,A.2}$ the proxy can fulfill not only the re-encryption from $A.1$ to $A.2$, but also the other way round. Here the number of re-encryption keys delivered in the communication channel could be reduced. If there are N employees in the group, only $N-1$ (i.e. $O(N)$) other than $N \times (N-1)$ (i.e. $O(N^2)$) re-encryption keys are required to be constructed and stored in the cloud. Nevertheless, none of existing MH-IBPRE supports the bidirectional property to date.

Open problems. The existing MH-IBPRE schemes leave us two interesting open problems: one is the fine-grained control of re-encryption, and the other is the construction of MH-IBPRE with constant complexity in terms of computation and communication. To tackle the problems, a novel MH-IBPRE system is desirable. The new scheme should not only support the bidirectional property, but also enjoy constant-size ciphertexts and decryption complexity no matter how many re-encryption hops have been traversed. Furthermore, it should allow the re-encryption power of the proxy to be limited to some pieces of conditions on the re-encryption key specified by the delegator (i.e. supporting *conditional re-encryption*).

1.1. Our contributions

- This paper formalizes the definition and security notion for bidirectional multi-hop identity-based conditional proxy re-encryption (BiMH-IBCPRE).
  1. In the definition, a condition is required as an auxiliary input to the re-encryption key generation, encryption, re-encryption and decryption algorithms.

[1] https://www.sugarsync.com/.
[2] https://www.box.com/.

2. In the security notion, this paper proposes the game-based definition of *selective condition and selective identity and adaptive chosen-ciphertext* (IND-sCon-sID-CCA) security, in which an adversary is restricted to output the challenge condition and the challenge identity before seeing the public parameters. Meanwhile, this paper proposes the definition of *collusion resistance* for BiMH-IBCPRE, in which the proxy cannot compromise the entire private key of the delegator even colluding with the corresponding delegatee.

- This paper, for the first time, proposes a BiMH-IBCPRE scheme to tackle the open problems left by the existing MH-IBPRE schemes. The proposed scheme is a type of MH-IBPRE, but additionally captures bidirectional, conditional re-encryption, constant decryption complexity and constant-size ciphertext properties. Note there are subtleties in combining bidirectional property, conditional re-encryption with MH-IBPRE to achieve a secure scheme, therefore our construction is not trivial. The paper also presents an extension of the new scheme which can support conditional re-encryption with a set of conditions.
- Our scheme can be proved IND-sCon-sID-CCA secure and collusion resistant in the standard model under the decisional 3-weak bilinear Diffie–Hellman inversion assumption. It is the first bidirectional PRE (in general) achieving collusion resistance.
- When compared with all existing MH-IBPRE schemes constructed in the standard model, our scheme achieves better performance in efficiency, and offers fine-grained decryption rights delegation to delegator without degrading security level.

1.2. Paper organization

The rest of this paper is organized as follows. In Section 2, we present the related work and the difficulty of achieving collusion resistance in bidirectional setting. In Section 3, we give the definitions and security models for BiMH-IBCPRE. In Section 4, we introduce some basic knowledge of bilinear map, complexity assumption, one-time signature and pseudorandom function family that will be used throughout the paper. In Section 5, we propose our basic BiMH-IBCPRE scheme in the standard model, and meanwhile, give the corresponding security proof. In Section 6, we give an efficient extension of our basic scheme. In Section 7, we compare our scheme with some well-known MH-IBPRE systems. In Section 8, we explore the applications of BiMH-IBCPRE. Finally, Section 9 concludes the paper. Appendix A presents the security proof of our basic scheme.

2. Related work

Following the concept of decryption rights delegation introduced by Mambo and Okamoto [22], Blaze, Bleumer and Strauss [3] formalized the notion of PRE, and proposed a seminal bidirectional PRE scheme with CPA security. In 2003, Ivan and Dodis [18] proposed formal definitions of the bidirectional and unidirectional proxy functions, and implemented the functions based on cryptographic primitives such as IBE. Later on, Ateniese et al. [1] proposed three CPA-secure unidirectional PRE schemes and demonstrated some practical applications for PRE. After that, many well-known unidirectional PRE schemes (e.g., [15,20,35,16,17]) have been proposed.

In 2007, Canetti and Hohenberger [7] defined the CCA security model for PRE, and proposed two CCA-secure bidirectional multi-hop PRE schemes with constant-size ciphertexts: one is constructed in the random oracle model, and the other is built in the standard model. Later on, Deng et al. [11] constructed a CCA-secure bidirectional single-hop PRE scheme in the random oracle model. In 2010, Matsuda, Nishimaki and Tanaka [23] proposed a bidirectional multi-hop PRE scheme that has been pointed out to be not CCA secure by Weng, Zhao and Hanaoka [36]. The aforementioned bidirectional PRE schemes, however, suffer from collusion attacks where a dishonest proxy can compromise the entire private key of the delegator by colluding with the corresponding delegatee. Therefore, how to be collusion resistant in the bidirectional setting remains elusive.

To employ PRE in the context of IBE (i.e. supporting identity-based re-encryption), Green and Ateniese [14] defined the notion of IBPRE and proposed two concrete constructions in the random oracle model: one is CCA-secure unidirectional single-hop, and the other is unidirectional multi-hop with CPA security. Later on, Tang, Hartel and Jonker [29] proposed a CPA-secure IBPRE scheme with random oracles, in which the delegator and the delegatee can belong to different domains. The previous schemes, however, are not collusion resistant. To hold against collusion attacks, the following unidirectional single-hop IBPRE schemes are proposed. Two CPA-secure IBPRE schemes without random oracles were proposed by Matsuo [24]. In 2010, Wang et al. [31,32] proposed two IBPRE schemes in the random oracle model: one is CPA-secure supporting the revocability of proxy's re-encryption rights, and the other is CCA-secure allowing the proxy to be malicious (rather than being semi-trusted). In 2011, Mizuno and Doi [25] constructed an IBPRE scheme in the standard model with CPA security based on Waters IBE [33] and an ElGamal-type PKE scheme. Later, two CPA-secure IBPRE schemes without random oracles were proposed by Luo, Shen and Chen [21]: one is single-hop and the other is multi-hop. Recently, Liang et al. [19] proposed the first CCA-secure unidirectional single-hop IBPRE in the standard model supporting conditional re-encryption. The scheme can be regarded as the combination of IBPRE and conditional PRE. Note that conditional PRE will be further introduced later.

Table 1
Functionality and security comparison.

Schemes	[7]	[14]	[9]	[27]	Ours
Security	CCA	CPA	RCCA	CCA	CCA
Without RO	✓	×	✓	✓	✓
Direction	↔	→	→	→	↔
Conditional re-encryption	×	×	×	×	✓
Constant size ciphertext	✓	×	×	×	✓
Collusion resistance	×	×	×	✓	✓

In [14], Green and Ateniese left the construction of a CCA-secure unidirectional MH-IBPRE as an open problem. To solve the problem, Chu and Tzeng [9] proposed an MH-IBPRE scheme without random oracles which is secure against replayable chosen-ciphertext attacks (RCCA) [8]. In 2010, Wang et al. [30] proposed the first CCA-secure MH-IBPRE with random oracles. Nevertheless, these schemes all suffer from collusion attacks. Recently, Shao and Cao [27] proposed a generic construction for CCA-secure unidirectional MH-IBPRE in the standard model, and meanwhile presented a concrete construction based on a fully secure IBE scheme [34]. The construction, however, has to depend on a hierarchical IBE that is IND-aID-CCA secure. Previously introduced MH-IBPRE schemes incur high communicational and computational complexity as the size of ciphertext and the decryption cost grow linearly in the number of re-encryption hops. This paper focuses on solving such a problem.

The above mentioned PREs (in general) cannot support fine-grained control on re-encryption such that the proxy can re-encrypt all ciphertexts of delegator to delegatee without any discrimination [37]. To address the problem in traditional PRE setting, Type-Based PRE (TBPRE) [28] and Conditional PRE (CPRE) [38] were proposed, in which the proxy can re-encrypt a ciphertext tagged with a condition if and only if the re-encryption key corresponding to the same condition is given by the delegator. In 2009, Weng et al. [37] proposed a new CPRE scheme by re-formalizing the definitions and security notions, and pointed out the secure risk of [38]. To hide the condition from proxy, Fang, Susilo and Wang [12] proposed a CCA-secure CPRE with condition anonymity in the standard model. Despite there are a few PRE schemes supporting conditional re-encryption property, how to achieve this property in the MH-IBPRE setting remains open. This paper also deals with the problem.

Difficulty of being secure against collusion attacks in bidirectional setting. In traditional PRE with bidirectional property, such as [7,11], the re-encryption key algorithm takes the private keys of delegator and delegatee as input, and outputs a re-encryption key, denoted as *rekey*. Due to the construction of *rekey* (e.g., $rekey = x/y$, x and y are the private keys of delegator and delegatee), the proxy can compromise the private key of the delegator by colluding with the delegatee. Thus, how to hold against collusion attacks in the bidirectional setting that remains open.

Intuitively, one might think that by using a random value, e.g., a β, to randomize the delegator's private key such that $rekey = (\beta \cdot x)/y$, the problem can be solved. However, this is an undesirable approach. We use Canetti and Hohenberger's bidirectional PRE scheme (the one with random oracles) [7] as an example to give a specific explanation. Note that for convenience we only analyze the components of ciphertext to be participated into re-encryption and decryption. In [7], the components of an original ciphertext are $A = svk$, $B = pk^r$, $C = e(g, H(svk))^r \cdot m$, where $pk = g^x$ $(sk = x)$, $g \in \mathbb{G}$, $x, r \in_R \mathbb{Z}_q^*$, H is a target collusion resistant hash function, m is the plaintext and svk is the verification key of a one time signature scheme. Given a $rekey = x/y$ (where y is the private key of the corresponding delegatee), the proxy outputs $B' = B^{y/x} = (g^{xr})^{y/x} = g^{ry}$. In decryption, the delegatee with y can recover m as $m = C/e(B', H(A))^{1/y} = e(g, H(svk))^r \cdot m/e(g^{ry}, H(svk))^{1/y}$. Here if we use β to randomize x such that $rekey = (\beta \cdot x)/y$, then the output of the re-encryption will become $B' = B^{y/\beta \cdot x} = (g^{xr})^{y/\beta \cdot x} = g^{yr/\beta}$.

It is clear that the delegatee (with knowledge of y) cannot recover the plaintext without knowledge of β. Hence, when generating *rekey* the delegator might choose to encrypt β under the public key of the delegatee such that the delegatee can recover m. If so, the bidirectional property cannot be maintained any more as $rekey = ((\beta \cdot x)/y, Enc(pk_y, \beta))$ is only applicable to the re-encryption from the delegator to the delegatee but not the inverse case. In addition, by colluding with the delegatee, the proxy still can compromise x with knowledge of y and β.

Alternatively, the delegator might hide β from the proxy and the delegatee by letting it be an exponent of its public key, that is, $pk = g^{\beta \cdot x}$ $(sk = x)$. Here B will become $B = (g^{\beta \cdot x})^r$. Then the proxy will output $B' = B^{y/(\beta \cdot x)} = (g^{(\beta \cdot x)r})^{y/(\beta \cdot x)} = g^{ry}$ such that the delegatee can recover m with y. It seems that the bidirectional property and collusion resistance can be obtained simultaneously as the proxy is only able to compromise $\beta \cdot x$ rather than x by collusion. This approach, however, comes at a price that the size of public key is linear in the number of re-encryption keys. For example, if there are N distinct re-encryption keys of a user to be uploaded to the proxy, then the user has to generate N corresponding public keys. Therefore, the above approach does not scale well in practice.

We compare our scheme with four multi-hop PRE schemes in terms of functionality and security in Table 1. Despite identity-based re-encryption, conditional re-encryption, collusion resistance and constant size ciphertext properties have all four been partially achieved by existing schemes, there is no efficient CCA-secure proposal that achieves the properties simultaneously in the standard model. This paper is the first to achieve the goal.

3. Definition and security models

3.1. Definition of BiMH-IBCPRE

Definition 1. A Bidirectional Multi-Hop Identity-Based Conditional Proxy Re-Encryption (BiMH-IBCPRE) scheme consists of the following six Probabilistic Polynomial Time (PPT) algorithms and protocol:

1. $(mpk, msk) \leftarrow Setup(1^\lambda, n)$: on input a security parameter $\lambda \in \mathbb{N}$ and $n \in \mathbb{N}$ the allowable maximum number of condition in the system, the algorithm *Setup* outputs a master public key *mpk* and a master secret key *msk* for Private Key Generator (PKG).
2. $sk_{ID} \leftarrow KeyGen(mpk, msk, ID)$: on input *mpk*, *msk*, and an identity $ID \in \{0,1\}^*$, the key generation algorithm *KeyGen* outputs a private key sk_{ID} for the identity *ID*.
3. Re-Encryption Key Generation Protocol: for simplicity, hereafter we use i and j to denote ID_i and ID_j in the re-encryption key.
   (a) $prk_{(W,j)} \leftarrow PReKeyGen(mpk, sk_{ID_j}, W)$: on input *mpk*, a private key sk_{ID_j} for identity ID_j, and a condition set $W = \{w_z \mid 1 \le z \le n, w_z \in \{0,1\}^*\}$, the partial re-encryption key generation algorithm *PReKeyGen* outputs a partial re-encryption key $prk_{(W,j)}$ under W and ID_j.
   (b) $rk_{i\to j|W} \leftarrow ReKeyGen(mpk, sk_{ID_i}, prk_{(W,j)}, W)$: on input *mpk*, a private key sk_{ID_i} for identity ID_i, a partial re-encryption key $prk_{(W,j)}$ and a set W of conditions, the re-encryption key algorithm *ReKeyGen* outputs a re-encryption key $rk_{i\to j|W}$ from ID_i to ID_j under W. Note ID_i and ID_j are two distinct identities.
   (c) $rk_{j\to i|W} \leftarrow ReKeyBiGen(rk_{i\to j|W})$: on input a re-encryption key $rk_{i\to j|W}$ from an identity ID_i to another identity ID_j under a set W of conditions, the re-encryption key derivation algorithm outputs a new re-encryption key $rk_{j\to i|W}$ from ID_j to ID_i under W. Note that this algorithm also allows the re-encryption key $rk_{j\to i|W}$ holder to generate a new re-encryption key $rk_{i\to j|W}$.
4. $C_{(ID_i,W)} \leftarrow Enc(mpk, ID_i, W, m)$: on input *mpk*, an identity ID_i, a set W of conditions and a message $m \in \{0,1\}^\lambda$, the encryption algorithm *Enc* outputs a first-level ciphertext (i.e. original ciphertext) $C_{(ID_i,W)}$ under ID_i and W. Note that ID_i and W are implicitly included in the ciphertext.
5. $C_{(ID_j,W)} \leftarrow ReEnc(mpk, rk_{i\to j|W}, C_{(ID_i,W)})$: on input *mpk*, a re-encryption key $rk_{i\to j|W}$, and a ciphertext $C_{(ID_i,W)}$, the re-encryption algorithm *ReEnc* outputs a re-encrypted ciphertext $C_{(ID_j,W)}$ or a symbol $\perp$ indicating that the ciphertext $C_{(ID_i,W)}$ is invalid.
6. $m \leftarrow Dec(mpk, sk_{ID_i}, C_{(ID_i,W)})$: on input *mpk*, a private key sk_{ID_i} for identity ID_i, and a ciphertext $C_{(ID_i,W)}$, the decryption algorithm *Dec* outputs a message m or a symbol $\perp$ indicating that the ciphertext $C_{(ID_i,W)}$ is invalid.

For simplicity, we omit *mpk* in the expression of the algorithms input in the rest of the paper.

Correctness: For any security parameter $\lambda \in \mathbb{N}$, any $n \in \mathbb{N}$, any identities $\{ID_i \in \{0,1\}^* \mid 1 \le i \le l, l = poly(1^\lambda)\}$, any condition set $W = \{w_z \mid 1 \le z \le n, w_z \in \{0,1\}^*\}$ and any message $m \in \{0,1\}^\lambda$, if $(mpk, msk) \leftarrow Setup(1^\lambda, n)$, $sk_{ID} \leftarrow KeyGen(msk, ID)$ (for all *ID* used in the system), $rk_{i\to i+1|W} \leftarrow ReKeyGen(sk_{ID_i}, prk_{(W,i+1)}, W)$, and $prk_{(W,i+1)} \leftarrow PReKey\text{-}Gen(sk_{ID_{i+1}}, W)$, we have $\forall i \in \{1, ..., l\}, m \leftarrow Dec(sk_{ID_i}, Enc(ID_i, W, m))$ and

1. Correctness for the multi-hop property:

$$m \leftarrow Dec\big(sk_{ID_l}, ReEnc\big(rk_{l-1\to l|W}, ReEnc\big(rk_{l-2\to l-1|W}, ..., ReEnc\big(rk_{1\to 2|W}, Enc(ID_1, W, m)\big)\big)\ldots\big)\big).$$

2. Correctness for the bidirectional property: $\forall i, j \in \{1, ..., l\}$, $i \neq j$,

$$m \leftarrow Dec\big(sk_{ID_j}, ReEnc\big(rk_{i\to j|W}, Enc(ID_i, W, m)\big)\big);$$
$$m \leftarrow Dec\big(sk_{ID_i}, ReEnc\big(ReKeyBiGen(rk_{i\to j|W}), Enc(ID_j, W, m)\big)\big),$$

   where the re-encryption key $rk_{j\to i|W} \leftarrow ReKeyBiGen(rk_{i\to j|W})$ (resp. $rk_{i\to j|W} \leftarrow ReKeyBiGen(rk_{j\to i|W})$[3]). Note that given $rk_{i\to j|W}$ (resp. $rk_{j\to i|W}$) the proxy can fulfill a bidirectional re-encryption between two distinct identities ID_i and ID_j under a set W of conditions with the help of algorithm *ReKeyBiGen*.

3.2. Security models

In this section we concentrate on formulating the security notion for BiMH-IBCPRE including IND-sCon-sID-CCA security and collusion resistance. Before proceeding, we define the notations that to be used in the definition as follows.

[3] In this paper we let $rk_{j\to i|W} = (rk_{i\to j|W})^{-1}$.

- **Delegation chain.** Suppose in a BiMH-IBCPRE scheme there is a re-encryption key set $RK = \{rk_{i_1 \to i_2|W}, ..., rk_{i_{l-1} \to i_l|W}\}$ $(l \geq 2)$, under the same condition set $W = \{w_z \mid 1 \leq i \leq n, w_z \in \{0,1\}^*\}$, for any re-encryption key $rk_{i_j \to i_{j+1}|W}$ in RK, $i_j \neq i_{j+1}$. We say that there exists a delegation chain under W from identity ID_{i_1} to identity ID_{i_l}, denoted as $(W, ID_{i_1} \to ... \to ID_{i_l})$. Due to the bidirectional property, we can achieve another re-encryption key set $RK^{BiDerive} = \{rk_{i_l \to i_{l-1}|W}, ..., rk_{i_2 \to i_1|W}\}$ $(l \geq 2)$, under W, which can be denoted as $(W, ID_{i_l} \to ... \to ID_{i_1})$. Therefore, given RK we have a delegation chain $(W, ID_{i_1} \leftrightarrow ... \leftrightarrow ID_{i_l})$ in the BiMH-IBCPRE scheme.
- **Uncorrupted/corrupted identity.** If the private key of an identity is compromised by an adversary, then the identity is considered as a corrupted identity. Otherwise, the identity is an uncorrupted identity.
- **Uncorrupted delegation chain.** Suppose there is a delegation chain under a set W of conditions between identity ID_i and identity ID_j (i.e. $(W, ID_i \leftrightarrow ... \leftrightarrow ID_j)$). If there is no corrupted identity on the chain, then it is an uncorrupted delegation chain. Otherwise, it is a corrupted one.

Definition 2. A BiMH-IBCPRE scheme is IND-sCon-sID-CCA secure if there is no PPT adversary $\mathcal{A}$ which can win the game below with non-negligible advantage. In the game, $\mathcal{C}$ is the challenger whom plays the game with the adversary $\mathcal{A}$, λ is the security parameter and n is the allowable maximum number of condition in the system.

1. **Init.** $\mathcal{A}$ outputs a challenge identity ID^* and a challenge set W^* of conditions to $\mathcal{C}$.
2. **Setup.** $\mathcal{C}$ runs $Setup(1^\lambda, n)$ and sends mpk to $\mathcal{A}$.
3. **Phase 1.** $\mathcal{A}$ is given access to the following oracles.
   (a) Private key extraction oracle $\mathcal{O}_{sk}(ID)$: on input an identity ID, $\mathcal{C}$ returns $sk_{ID} \leftarrow KeyGen(msk, ID)$.
   (b) Re-encryption key extraction oracle $\mathcal{O}_{rk}(ID_i, ID_j, W)$: on input two distinct identities ID_i and ID_j, and a condition set W, $\mathcal{C}$ returns a re-encryption key $rk_{i \to j|W} \leftarrow ReKeyGen(sk_{ID_i}, PReKeyGen(sk_{ID_j}, W), W)$, where $sk_{ID_i} \leftarrow KeyGen(msk, ID_i)$, $sk_{ID_j} \leftarrow KeyGen(msk, ID_j)$, and $ID_i, ID_j \in \{0,1\}^*$. Note that $\mathcal{A}$ can derive $rk_{j \to i|W}$ from $rk_{i \to j|W}$ with algorithm $ReKeyBiGen$.
   (c) Re-encryption oracle $\mathcal{O}_{re}(ID_i, ID_j, W, C_{(ID_i,W)})$: on input two distinct identities ID_i and ID_j, a condition set W, and a ciphertext $C_{(ID_i,W)}$ under ID_i and W, $\mathcal{C}$ returns a re-encrypted ciphertext $C_{(ID_j,W)} \leftarrow ReEnc(rk_{i \to j|W}, C_{(ID_i,W)})$, where $rk_{i \to j|W} \leftarrow ReKeyGen(sk_{ID_i}, PReKeyGen(sk_{ID_j}, W), W)$, $sk_{ID_i} \leftarrow KeyGen(msk, ID_i)$, $sk_{ID_j} \leftarrow KeyGen(msk, ID_j)$, ID_i, $ID_j \in \{0,1\}^*$. Note that it could be possible for $\mathcal{A}$ to issue the query $(ID_j, ID_i, W, C_{(ID_i,W)})$ to $\mathcal{O}_{re}$. If so, $\mathcal{C}$ will first generate $rk_{j \to i|W}$ and get $rk_{i \to j|W} \leftarrow ReKeyBiGen(rk_{j \to i|W})$, and further re-encrypt $C_{(ID_i,W)}$ using $rk_{i \to j|W}$.
   (d) Decryption oracle $\mathcal{O}_{dec}(ID_i, C_{(ID_i,W)})$: on input an identity ID_i, and a ciphertext $C_{(ID_i,W)}$, $\mathcal{C}$ returns $m \leftarrow Dec(sk_{ID_i}, C_{(ID_i,W)})$, where $sk_{ID_i} \leftarrow KeyGen(msk, ID_i)$, $ID_i \in \{0,1\}^*$.
   Note that if $\mathcal{A}$ issues invalid ciphertexts to $\mathcal{O}_{re}$ or $\mathcal{O}_{dec}$, $\mathcal{C}$ simply outputs $\perp$. Moreover, the following queries cannot be issued:
   - $\mathcal{O}_{sk}(ID)$ if $ID^* = ID$ or for any ID in an uncorrupted delegation chain under W^* which includes ID^*;
   - $\mathcal{O}_{rk}(ID_i, ID_j, W^*)$ for any distinct ID_i and ID_j, if ID^* will be in a corrupted delegation chain under W^* after issuing the corresponding re-encryption key.
4. **Challenge.** $\mathcal{A}$ outputs two distinct equal length messages (m_0, m_1) to $\mathcal{C}$. $\mathcal{C}$ returns the challenge ciphertext $C^*_{(ID^*,W^*)} = Enc(ID^*, W^*, m_b)$ to $\mathcal{A}$, where $b \in_R \{0,1\}$.
5. **Phase 2.** $\mathcal{A}$ continues making queries except the followings:
   (a) $\mathcal{O}_{sk}(ID)$ if $ID^* = ID$ or for any ID in an uncorrupted delegation chain under W^* which includes ID^*;
   (b) $\mathcal{O}_{rk}(ID_i, ID_j, W^*)$ for any distinct ID_i and ID_j, if ID^* will be in a corrupted delegation chain under W^* after issuing the corresponding re-encryption key.
   (c) $\mathcal{O}_{re}(ID_i, ID_j, W^*, C_{(ID_i,W^*)})$ if $(ID_i, W^*, C_{(ID_i,W^*)})$ is a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$, but ID_j is a corrupted identity or ID_j is in a corrupted delegation chain.[4] As of [7], a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$ is defined as follows.
      i. $(ID^*, W^*, C^*_{(ID^*,W^*)})$ is a derivative of itself.
      ii. If $(ID_i, W^*, C_{(ID_i,W^*)})$ is a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$, $(ID_{i'}, W^*, C_{(ID_{i'},W^*)})$ is a derivative of $(ID_i, W^*, C_{(ID_i,W^*)})$, then $(ID_{i'}, W^*, C_{(ID_{i'},W^*)})$ is a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$.
      iii. If $\mathcal{A}$ has issued a re-encryption key query on (ID_i, ID_j, W) to obtain $rk_{i \to j|W}$, and achieved $C_{(ID_j,W)} \leftarrow ReEnc(rk_{i \to j|W}, C_{(ID_i,W)})$, then $(ID_j, W, C_{(ID_j,W)})$ is a derivative of $(ID_i, W, C_{(ID_i,W)})$.
      iv. If $\mathcal{A}$ can run $C_{(ID_j,W)} \leftarrow ReEnc(ReKeyGen(sk_{ID_i}, prk_{(W,j)}, W), C_{(ID_i,W)})$, then $(ID_j, W, C_{(ID_j,W)})$ is a derivative of $(ID_i, W, C_{(ID_i,W)})$, where $sk_{ID_i} \leftarrow KeyGen(msk, ID_i)$, $prk_{(W,j)} \leftarrow PReKeyGen(sk_{ID_j}, W)$ and $sk_{ID_j} \leftarrow KeyGen(msk, ID_j)$.
      v. If $\mathcal{A}$ has issued a re-encryption query on $(ID_i, ID_j, W, C_{(ID_i,W)})$ and obtained $C_{(ID_j,W)}$, then $(ID_j, W, C_{(ID_j,W)})$ is a derivative of $(ID_i, W, C_{(ID_i,W)})$.
   (d) $\mathcal{O}_{dec}(ID_i, C_{(ID_i,W^*)})$ if $(ID_i, W^*, C_{(ID_i,W^*)})$ is a derivative of $(ID^*, W^*, C^*_{(ID^*,W^*)})$.
6. **Guess.** $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$. If $b' = b$, $\mathcal{A}$ wins.

The advantage of $\mathcal{A}$ is defined as $\epsilon_1 = Adv^{\text{IND-sCon-sID-CCA}}_{\text{BiMH-IBCPRE},\mathcal{A}}(1^\lambda, n) = |Pr[b' = b] - \frac{1}{2}|$.

[4] Namely, ID^* is in a corrupted delegation chain under W^* after re-encryption.

Remarks. When $W = null$, the model above is for BiMH-IBPRE. The model will become IND-sCon-sID-CPA if $\mathcal{A}$ cannot query $\mathcal{O}_{re}$ and $\mathcal{O}_{dec}$.

Usually a traditional PRE has two types of security notion: one is for the original ciphertext security, and the other is for the re-encrypted ciphertext security (which usually implies collusion resistance). From our security model, it can be seen that we only define the first type of security notion, i.e. the security of original ciphertext (as the challenge is an original ciphertext). We state that the security model already implies the security for re-encrypted ciphertext in the BiMH-IBCPRE setting with a limitation that the challenge identity must be in an uncorrupted delegation chain under the challenge condition.[5] The above statement is valid because of the bidirectional and multi-hop properties of a BiMH-IBCPRE. Specifically, if given an l-th level (i.e. being re-encrypted $l-1$ times) re-encrypted ciphertext under identity ID_j and challenge condition W^* as challenge, $\mathcal{A}$ can convert the ciphertext to the original (1-th) one under identity ID_i and W^*, i.e. the original ciphertext without any re-encryption. This can be done as the uncorrupted delegation chain under W^* between ID_j and ID_i is allowed to be issued by $\mathcal{A}$ in the our model. Similarly, given an original ciphertext under ID_i and W^* as challenge, $\mathcal{A}$ also can construct an l-th re-encrypted ciphertext under ID_j and W^* with knowledge of the delegation chain. Therefore, it makes no difference for $\mathcal{A}$ to be given either an original ciphertext or a re-encrypted one as challenge.

In Definition 2, we do not allow $\mathcal{A}$ to issue the queries of partial re-encryption key. The reason is that we assume there is a secure channel between delegator and delegatee, such as SSL, for delivering partial re-encryption key in our system. However, this assumption does not indicate that $\mathcal{A}$ cannot obtain any partial re-encryption key. For instance, there may be a re-encryption key from ID_i (corrupted by $\mathcal{A}$) to ID_j under a condition set W' (which is not the challenge W^*). With knowledge of $rk_{i\to j|W}$ and sk_{ID_i}, $\mathcal{A}$ may recover a partial re-encryption key $prk_{(W,j)}$. This can be seen in our construction (please refer to Section 5).

Note that a stronger security notion, i.e. allowing $\mathcal{A}$ to issue the partial re-encryption key queries, can be extended from our model by giving $\mathcal{A}$ an additional oracle $\mathcal{O}_{prk}$ which takes an identity ID and a condition set W as input and outputs a partial re-encryption key. Nevertheless, $\mathcal{O}_{prk}$ must follow a restriction where if an identity ID is in a delegation chain under W^* containing the challenge identity ID^*, then the oracle outputs $\perp$. This is necessary because if such a partial re-encryption key is given, $\mathcal{A}$ can generate a re-encryption key from ID to a corrupted identity such that ID^* is in a corrupted delegation chain under W^*. This contradicts our IND-sCon-sID-CCA game.

We next proceed to the definition of collusion resistance. Since the security model defined in Definition 2 is based on the selective model (i.e. sCon-sID), we define the collusion resistance for BiMH-IBCPRE below in the selective model as well.

Definition 3. A BiMH-IBCPRE scheme is collusion resistant if the advantage $Adv^{CR}_{\mathcal{A}}(1^\lambda, n)$ is negligible for any PPT adversary $\mathcal{A}$ in the following experiment. Set $O = \{\mathcal{O}_{sk}, \mathcal{O}_{rk}\}$.

$$Adv^{CR}_{\mathcal{A}}(1^\lambda, n) = Pr\left[sk_{ID^*} \in \Omega : (ID^*, W^*, State) \leftarrow \mathcal{A}(1^\lambda), (mpk, msk) \leftarrow Setup(1^\lambda, n); sk_{ID^*} \leftarrow \mathcal{A}^O(mpk, State)\right]$$

where λ is the security parameter, ID^* and W^* are the challenge identity and the challenge condition set, Ω is the valid private key space, $\mathcal{O}_{sk}$ and $\mathcal{O}_{rk}$ are the oracles defined in Definition 2, but there is no constraint for querying them except for $\mathcal{O}_{sk}(ID^*)$.

Below we show that the IND-sCon-sID-CCA security already implies collusion resistance.

Theorem 1. *If a BiMH-IBCPRE scheme achieves IND-sCon-sID-CCA security as defined in Definition 2, it also achieves collusion resistance.*

Proof. In the game of Definition 2, an adversary $\mathcal{A}$ is allowed to access the re-encryption keys $rk_{i^*\to j|W^*}$ and $rk_{j\to l|W}$, where $W \neq W^*$, W^* is the challenge condition set, ID^*_i is the challenge identity, ID_j is an uncorrupted identity and ID_l is corrupted by $\mathcal{A}$.

Suppose a BiMH-IBCPRE scheme is not collusion resistant, that is, given sk_{ID_l} and $rk_{j\to l|W}$, $\mathcal{A}$ can compromise the private key sk_{ID_j}. After receiving the challenge ciphertext $C^*_{(ID^*,W^*)}$ from the challenger $\mathcal{C}$, $\mathcal{A}$ can re-encrypt the ciphertext by using $rk_{i^*\to j|W^*}$ to obtain the re-encrypted ciphertext $C^*_{(ID_j,W^*)}$. $\mathcal{A}$ further decrypts $C^*_{(ID_j,W^*)}$ by using sk_{ID_j} so as to know the value of the bit b. Accordingly, the security of IND-sCon-sID-CCA fails that contradicts our security notion.

Therefore, the IND-sCon-sID-CCA security implies collusion resistance.

This completes the proof of Theorem 1. □

4. Preliminaries

Below we review negligible function, bilinear maps and the complexity assumption used in our security proof, and next introduce strongly existential unforgeable one-time signature and the pseudorandom function family.

[5] This must be followed in multi-hop setting. This is so because if the adversary $\mathcal{A}$ obtains a delegation chain under the challenge condition containing the challenge identity and a corrupted identity (in the chain), $\mathcal{A}$ (with knowledge of the delegation chain) can re-encrypt the challenge ciphertext to another ciphertext which can be decrypted by using the private key sk_{ID_j}, where ID_j is corrupted.

4.1. Negligible function

A function $\phi(n) : N \to R$ is negligible in n if $1/\phi(n)$ is a non-polynomially-bounded quantity in n.

4.2. Bilinear maps

Let *BSetup* be an algorithm that on input the security parameter λ, outputs the parameters of a bilinear map as $(q, g, \mathbb{G}_1, \mathbb{G}_T, e)$, where $\mathbb{G}_1$ and $\mathbb{G}_T$ are multiplicative cyclic groups of prime order q, where $|q| = \lambda$, and g is a random generator of $\mathbb{G}_1$. The mapping $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_T$ has three properties: (1) *Bilinearity*: for all $a, b \in_R \mathbb{Z}_q^*$, $e(g^a, g^b) = e(g, g)^{ab}$; (2) *Non-degeneracy*: $e(g, g) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the unit of $\mathbb{G}_T$; (3) *Computability*: e can be efficiently computed.

4.3. The decisional l-th weak bilinear Diffie–Hellman inversion assumption

Definition 4. Given the tuple $(g, h, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^l}, T) \in \mathbb{G}_1^{l+2} \times \mathbb{G}_T$, the decisional l-wBDHI* problem is to decide whether $T = e(g, h)^{\alpha^{l+1}}$, where h is a generator of $\mathbb{G}_1$ and $\alpha \in \mathbb{Z}_q^*$. Define

$$\left|Pr\left[\mathcal{A}\left(g, h, g^{\alpha}, ..., g^{\alpha^l}, e(g, h)^{\alpha^{l+1}}\right) = 0\right] - Pr\left[\mathcal{A}\left(g, h, g^{\alpha}, ..., g^{\alpha^l}, T\right) = 0\right]\right|$$

as the advantage $Adv_{\mathcal{A}}^{D\text{-}l\text{-}\mathrm{wBDHI}^*}$ of an adversary $\mathcal{A}$ in winning the decisional l-wBDHI* problem. We say that the decisional l-wBDHI* assumption [4] holds in $\mathbb{G}_1$ if no PPT $\mathcal{A}$ has non-negligible advantage.

4.4. Strongly existential unforgeable one-time signatures

A strongly existential unforgeable one-time signature (OTS) [2] consists of the following algorithms:

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

Definition 5. An OTS scheme is one-time strongly unforgeable chosen-message attack (sUF-CMA) secure if the advantage $Adv_{\mathcal{A}}^{\mathrm{OTS}}(1^{\lambda})$ is negligible for any PPT adversary $\mathcal{A}$ in the following experiment.

$$Adv_{\mathcal{A}}^{\mathrm{OTS}}\left(1^{\lambda}\right) = Pr\left[Verify\left(K_v, \sigma^*, M^*\right) = 1 : (K_s, K_v) \leftarrow Sign.KeyGen\left(1^{\lambda}\right);\right.$$
$$\left.(M, State) \leftarrow \mathcal{A}(K_v); \sigma \leftarrow Sign(K_s, M); \left(M^*, \sigma^*\right) \leftarrow \mathcal{A}(K_v, \sigma, State); \left(M^*, \sigma^*\right) \neq (M, \sigma)\right],$$

where *State* is the state information.

4.5. Pseudorandom function family

Let $PRF : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$ be a pseudorandom function family [13], and $G : \mathcal{D} \to \mathcal{R}$ be a truly random function family, where $\mathcal{K}$ is the set of function keys, $\mathcal{D}$ and $\mathcal{R}$ are the domain and range, respectively. In short, we use $PRF_k(m)$ to represent $PRF(k, m)$. The advantage of a PPT adversary $\mathcal{A}$ in attacking the pseudorandomness of PRF is defined as:

$$Adv_{\mathcal{A}}^{PRF}\left(1^{\lambda}\right) = \left|Pr\left[\mathcal{A}\left(m \in \mathcal{D}, PRF_k(m)\right) = 1 : k \overset{R}{\leftarrow} \mathcal{K}\right] - Pr\left[\mathcal{A}\left(m \in \mathcal{D}, G(m)\right) = 1\right]\right|,$$

where λ is a security parameter. If $Adv_{\mathcal{A}}^{PRF}(1^{\lambda})$ is negligible for any PPT adversary $\mathcal{A}$ in attacking the pseudorandomness of the function family PRF, then PRF is a pseudorandom function family.

5. A CCA-secure BiMH-IBCPRE scheme

In this section we first briefly introduce the roadmap of our construction, and then propose a concrete construction for CCA-secure BiMH-IBCPRE in the standard model. Meanwhile, we present the security proof right after the construction.

5.1. Basic construction

Construction intuition. The scheme is constructed based on a Hierarchical Identity-Based Encryption (HIBE) [4], a pseudorandom function [13], and a one-time signature scheme [2]. Our technical roadmap is different from that of [14,9,27]. In [14], Green and Ateniese began with Boneh and Franklin IBE [6] and next extended the IBE to a CPA-secure MH-IBPRE in the random oracle model. Chu and Tzeng [9] used the CHK transformation [5] to convert Waters IBE [33] to a CCA-secure 2-level IBE scheme, and then extended it to an RCCA-secure MH-IBPRE in the standard model. In [27], a CCA-secure MH-IBPRE without random oracles is built on top of a strongly CPA-secure HIBE scheme [34]. Note that [14,9,27] are constructed in the unidirectional setting.

However, we start with a 3-level HIBE scheme [4] such that an identity of a message receiver, a condition set associated with the message and a verification key can be respectively assigned to different level upon encrypting the message, and further extend the scheme to be secure against CCA by leveraging the CHK transformation. We next employ the technology of CPRE and a pseudorandom function in the CCA-secure 3-level HIBE scheme so as to propose an IBPRE with conditional re-encryption (i.e. CCA-secure IBCPRE). The usage of pseudorandom function is to guarantee the validity of re-encryption such that once the re-encryption result is mutated by an adversary, the mutation can be told. Note we will give more details in the security proof of our construction.

Furthermore, we adapt the CCA-secure IBCPRE to achieve the bidirectional property (i.e. CCA-secure Bi-IBCPRE) by revising the construction of re-encryption key. We allow a proxy to use a re-encryption key to partially "decrypt" an original ciphertext such that the delegatee can recover the plaintext by using his/her private key. The most subtle part is the generation of re-encryption key. The re-encryption key generation takes the private keys of a delegator and a delegatee as input. To preclude the proxy from accessing the underlying plaintext, we first use random factors to mask the private keys to become random elements, and next leverage the elements to construct the re-encryption key. Intuitively, these random elements will also mask the re-encryption result such that the delegatee cannot recover the plaintext unless he/she has knowledge of them (e.g., the elements are encrypted intended for the delegatee). But if the delegatee knows the random elements, then the proxy can easily compromise the entire private key of the delegator by colluding with the delegatee. Accordingly, the system cannot hold against collusion attacks. To address the problem, we make use of the property of bilinear pairing in the sense that we leverage random factors (which are unknown to both the proxy and the delegatee) to hide the private keys' components but these factors will be eliminated eventually in re-encryption phase.

Due to our re-encryption key generation technique, the re-encryption algorithm will always yield constant-size re-encrypted ciphertext no matter how many re-encryption hops have been traversed. This is so because there is no need to construct additional encryption for random factors in re-encryption key generation phase any more. Thus it is not difficult for us to extend the above resulting scheme to support multi-hop re-encryption.

Here we propose our basic construction for BiMH-IBCPRE where a ciphertext is only tagged with a single condition w, and in the next section (i.e. Section 6), we further extend the basic scheme to support multiple conditions which are denoted as a condition set $W = \{w_z \mid 1 \le z \le n, w_z \in \mathbb{Z}_q^*\}$. Denote by $[M]^{\lambda_1-\lambda}$ the first $\lambda_1 - \lambda$ bits of bit-string M, and $[M]_\lambda$ the last λ bits of bit-string M. Below is the description of our construction.

1. $Setup(1^\lambda, n)$. Given the security parameter λ and n the allowable maximum number of conditions in the system (here $n = 1$), run $(q, g, \mathbb{G}_1, \mathbb{G}_T, e) \leftarrow BSetup(1^\lambda)$, choose $\alpha \in_R \mathbb{Z}_q^*$, $f_1, f_2, g_2, g_3, h_1, h_2, h_3 \in_R \mathbb{G}_1$, and prepare a pseudorandom function $PRF : \mathbb{G}_T \times \mathbb{G}_1 \to \{0,1\}^{\lambda_1}$ (which takes an element in $\mathbb{G}_T$ as the function key and an element in $\mathbb{G}_1$ as input, and outputs a λ_1-bit pseudorandom string), where λ_1 is a security parameter as well. Let $w \in \mathbb{Z}_q^*$ be a condition, $(Sign.KeyGen, Sign, Verify)$ be an OTS scheme, and assume the verification key output by $Sign.KeyGen(1^\lambda)$ is in $\mathbb{Z}_q^*$. The master secret key is $msk = g_2^\alpha$, and the master public key is $mpk = (q, g, \mathbb{G}_1, \mathbb{G}_T, e, f_1, f_2, g_1, g_2, g_3, h_1, h_2, h_3, PRF, (Sign.KeyGen, Sign, Verify))$, where $g_1 = g^\alpha$.
2. $KeyGen(msk, ID)$. Given the master secret key msk and an identity ID (i.e. an identity $I = (ID)$), choose $r \in_R \mathbb{Z}_q^*$, and output the key
$$sk_{ID} = \left(g_2^\alpha \cdot \left(h_1^{ID} \cdot g_3\right)^r, g^r, h_2^r, h_3^r\right) \in \mathbb{G}_1^4.$$
A system user with knowledge of sk_{ID} can generate the following private key due to the key derivation of HIBE [4]. Given $sk_{ID} = (a_0, a_1, b_2, b_3)$, $I = (ID, w)$, choose a $t \in_R \mathbb{Z}_q^*$ and output
$$sk_{(ID,w)} = \left(a_0 \cdot b_2^w \cdot \left(h_1^{ID} \cdot h_2^w \cdot g_3\right)^t, a_1 \cdot g^t, b_3 \cdot h_3^t\right) = \left(g_2^\alpha \cdot \left(h_1^{ID} \cdot h_2^w \cdot g_3\right)^{r'}, g^{r'}, h_3^{r'}\right),$$
where $r' = r + t$.
3. $Enc(ID_i, w, m)$. Given an identity ID_i, a condition w and a message m, choose an OTS key pair $(K_s, K_v) \leftarrow Sign.KeyGen(1^\lambda)$ and $\sigma \in_R \mathbb{G}_T$, $s \in_R \mathbb{Z}_q^*$, set $C_0 = K_v$,
$$C_1 = \left[PRF_\sigma(C_3)\right]^{\lambda_1-\lambda} \left\| \left[PRF_\sigma(C_3)\right]_\lambda \oplus m, \qquad C_2 = \sigma \cdot e(g_1, g_2)^s,$$
$$C_3 = g^s, \qquad C_4 = \left(h_1^{ID_i} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3\right)^s,$$
$$C_5 = \left(f_1^w \cdot f_2\right)^s, \qquad C_6 = Sign\left(K_s, (C_1, C_3, C_4, C_5)\right),$$
and output the first-level ciphertext $C_{(ID_i,w)} = ((ID_i, w), C_0, C_1, C_2, C_3, C_4, C_5, C_6)$, where $ID_i \in \mathbb{Z}_q^*$, $m \in \{0,1\}^\lambda$.

4. Re-Encryption Key Generation Protocol.
   (a) $PReKeyGen(sk_{ID_j}, w)$. ID_j first deduces $sk_{(ID_j,w)} = (a_{0j}, a_{1j}, b_{3j})$ (under $I = (ID_j, w)$) from sk_{ID_j}, next chooses $\rho_1, \rho_2 \in_R \mathbb{Z}_q^*$ and sets $\beta_1 = {a_{0j}}^{-1} \cdot (f_1^w \cdot f_2)^{\rho_1}$, $\beta_2 = g^{\rho_1}$, $\beta_3 = {a_{1j}}^{-1} \cdot g^{\rho_2}$, $\beta_4 = b_{3j}^{-1} \cdot h_3^{\rho_2}$, $\beta_5 = (h_2^w \cdot g_3)^{\rho_2}$, $\beta_6 = h_1^{\rho_2}$. ID_j then sends the partial re-encryption key $prk_{(w,j)} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6)$ to ID_i.
   (b) $ReKeyGen(sk_{ID_i}, prk_{(w,j)}, w)$. ID_i first generates $sk_{(ID_i,w)} = (a_{0i}, a_{1i}, b_{3i})$ (under $I = (ID_i, w)$), chooses $\rho_3, \rho_4 \in_R \mathbb{Z}_q^*$, computes $rk_1 = a_{0i} \cdot \beta_1 \cdot (f_1^w \cdot f_2)^{\rho_3}$, $rk_2 = g^{\rho_3} \cdot \beta_2$, $rk_3 = a_{1i} \cdot \beta_3 \cdot g^{\rho_4}$, $rk_4 = b_{3i} \cdot \beta_4 \cdot h_3^{\rho_4}$, $rk_5 = (h_2^w \cdot g_3)^{\rho_4} \cdot \beta_5$, $rk_6 = h_1^{\rho_4} \cdot \beta_6$ and outputs $rk_{i\to j|w} = (rk_1, rk_2, rk_3, rk_4, rk_5, rk_6)$.
   (c) Derive a new re-encryption key $rk_{j\to i|w}$ from $rk_{i\to j|w}$ by running $rk_{j\to i|w} \leftarrow ReKeyBiGen(rk_{i\to j|w})$. The proxy sets $rk_{j\to i|w} = rk_{i\to j|w}^{-1}$, i.e. $rk_{j\to i|w} = (rk_1^{-1}, rk_2^{-1}, rk_3^{-1}, rk_4^{-1}, rk_5^{-1}, rk_6^{-1})$.
5. $ReEnc(rk_{i\to j|w}, C_{(ID_i,w)})$. Given a re-encryption key $rk_{i\to j|w}$ and a ciphertext $C_{(ID_i,w)}$, the re-encryption algorithm works as follows.
   (a) Verify the validity of the ciphertext:

$$e\big(C_3, f_1^w \cdot f_2\big) \stackrel{?}{=} e(g, C_5), \qquad e\big(C_3, h_1^{ID_i} \cdot h_2^w \cdot h_3^{C_0} \cdot g_3\big) \stackrel{?}{=} e(g, C_4),$$
$$Verify\big(C_0, C_6, (C_1, C_3, C_4, C_5)\big) \stackrel{?}{=} 1. \tag{1}$$

   If Eq. (1) does not hold, output $\perp$. Otherwise, proceed.
   (b) Compute $C_2^{(l)} = \frac{C_2^{(l-1)} \cdot e(rk_3, C_4) \cdot e(rk_2, C_5)}{e(rk_1 \cdot rk_4^{C_0} \cdot rk_6^{ID_i} \cdot rk_5, C_3)}$, output the re-encrypted ciphertext $C_{(ID_j,w)} = ((ID_i, w), C_0, C_1, C_2^{(l)}, C_3, C_4, C_5, C_6)$, where $l \geq 2$ denotes the level of the ciphertext. If $l = 1$, $C_2^{(1)}$ is from the first-level ciphertext.
6. $Dec(sk_{ID_i}, C_{(ID_i,w)})$. Given a private key sk_{ID_i} for ID_i and a ciphertext $C_{(ID_i,w)}$, the decryption algorithm works as follows. ID_i first deduces the private key $sk_{(ID_i,w)} = (a_{0i}, a_{1i}, b_{3i})$ (under $I = (ID_i, w)$) from sk_{ID_i} (under $I = (ID_i)$), and next does the followings.
   (a) Verify the validity of the ciphertext by checking Eq. (1). If the equation does not hold, output $\perp$. Otherwise, proceed.
   (b) Compute $\sigma = C_2^{(l)} \cdot \frac{e(a_{1i}, C_4)}{e(a_{0i} \cdot b_{3i}^{C_0}, C_3)}$, and then verify $[PRF_\sigma(C_3)]^{\lambda_1-\lambda} \stackrel{?}{=} [C_1]^{\lambda_1-\lambda}$. If the equation holds, output $m = [C_1]_\lambda \oplus [PRF_\sigma(C_3)]_\lambda$. Otherwise, output $\perp$.

Remarks. We assume there is a secure channel (such as SSL) between the delegator and the delegatee in the re-encryption key generation protocol. In addition, like [4], we can use a target collision resistant hash function [10]: $H : \{0,1\}^* \to \mathbb{Z}_q^*$ to hash each condition w, identity ID and verification key K_v beforehand such that we can extend our scheme to handle arbitrary-length condition, identity and verification key.

Correctness:

- When $l = 1$, $C_2^{(1)} = \sigma \cdot e(g_1, g_2)^s$. With knowledge of private key $sk_{(ID_i,w)} = (a_{0i}, a_{1i}, b_{3i}) = (g_2^\alpha \cdot (h_1^{ID_i} \cdot h_2^w \cdot g_3)^r, g^r, h_3^r)$, the original receiver can compute

$$\frac{C_2^{(1)} \cdot e(a_{1i}, C_4)}{e(a_{0i} \cdot b_{3i}^{C_0}, C_3)} = \frac{\sigma \cdot e(g_1, g_2)^s \cdot e(g^r, (h_1^{ID_i} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^s)}{e(g_2^\alpha \cdot (h_1^{ID_i} \cdot h_2^w \cdot g_3)^r \cdot h_3^{r \cdot K_v}, g^s)}$$
$$= \frac{\sigma \cdot e(g_1, g_2)^s}{e(g_2^\alpha, g^s)} = \frac{\sigma \cdot e(g_1, g_2)^s}{e(g_2, g_1^s)} = \sigma.$$

- When $l \geq 2$, suppose there is a delegation chain under w between ID_i and ID_j, i.e. $(w, ID_i \leftrightarrow \ldots \leftrightarrow ID_j)$, we have

$$C_2^{(2)} = \frac{C_2^{(1)} \cdot e(rk_3, C_4) \cdot e(rk_2, C_5)}{e(rk_1 \cdot rk_4^{C_0} \cdot rk_6^{ID_i} \cdot rk_5, C_3)}$$
$$= \frac{\sigma \cdot e(g_1, g_2)^s \cdot e(g^{r_i - r_{i+1}}, (h_1^{ID_i} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^s)}{e((h_1^{ID_i} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^{r_i} \cdot (h_1^{ID_{i+1}} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^{-r_{i+1}}, g^s)}$$
$$= \frac{\sigma \cdot e(g_1, g_2)^s \cdot e(g^{-r_{i+1}}, (h_1^{ID_i} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^s)}{e((h_1^{ID_{i+1}} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^{-r_{i+1}}, g^s)},$$
$$C_2^{(3)} = \frac{C_2^{(2)} \cdot e(rk'_3, C_4) \cdot e(rk'_2, C_5)}{e(rk'_1 \cdot rk_4'^{C_0} \cdot rk_6'^{ID_i} \cdot rk'_5, C_3)}$$
$$= \frac{\sigma \cdot e(g_1, g_2)^s \cdot e(g^{-r_{i+2}}, (h_1^{ID_i} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^s)}{e((h_1^{ID_{i+2}} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^{-r_{i+2}}, g^s)},$$

$$\vdots$$

$$C_2^{(j)} = \frac{C_2^{(j-1)} \cdot e(rk_3'', C_4) \cdot e(rk_2'', C_5)}{e(rk_1'' \cdot rk_4''^{C_0} \cdot rk_6''^{ID_i} \cdot rk_5'', C_3)}$$
$$= \frac{\sigma \cdot e(g_1, g_2)^s \cdot e(g^{-r_j}, (h_1^{ID_i} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^s)}{e((h_1^{ID_j} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^{-r_j}, g^s)}$$
$$= \frac{\sigma \cdot e(g_1, g_2)^s \cdot e(g^{-r_j}, C_4)}{e((h_1^{ID_j} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^{-r_j}, C_3)},$$

where $l = j$. With knowledge of private key $sk_{(ID_j, w)} = (a_{0j}, a_{1j}, b_{3j}) = (g_2^\alpha \cdot (h_1^{ID_j} \cdot h_2^w \cdot g_3)^{r_j}, g^{r_j}, h_3^{r_j})$, the delegatee ID_j can compute

$$\frac{\sigma \cdot e(g_1, g_2)^s \cdot e(g^{-r_j}, C_4) \cdot e(a_{1j}, C_4)}{e((h_1^{ID_j} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^{-r_j}, C_3) \cdot e(a_{0j} \cdot b_{3j}^{C_0}, C_3)}$$
$$= \frac{\sigma \cdot e(g_1, g_2)^s \cdot e(g^{-r_j}, C_4) \cdot e(g^{r_j}, C_4)}{e((h_1^{ID_j} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^{-r_j}, C_3) \cdot e(g_2^\alpha \cdot (h_1^{ID_j} \cdot h_2^w \cdot h_3^{K_v} \cdot g_3)^{r_j}, C_3)}$$
$$= \frac{\sigma \cdot e(g_1, g_2)^s}{e(g_2^\alpha, g^s)} = \frac{\sigma \cdot e(g_1, g_2)^s}{e(g_2, g_1^s)} = \sigma.$$

5.2. Security analysis

Below we prove our BiMH-IBCPRE scheme IND-sCon-sID-CCA secure in the standard model.

Informally, we state that our scheme is secure against CCA. Suppose there is an adversary $\mathcal{A}$ that follows the constraints defined in Definition 2. Let $C^*_{(ID^*, w^*)} = ((ID^*, w^*), C_0^*, C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$ be the challenge ciphertext. It is not difficult to see that the components C_1^*, C_3^*, C_4^* and C_5^* are bound by the signature C_6^*, the verification key C_0^*, ID^* and w^* are bound by C_4^*, and C_2^* is bound by the first $\lambda_1 - \lambda$ bits of C_1^* (i.e. $[PRF_\sigma(C_3^*)]^{\lambda_1 - \lambda}$). Without any trivial queries for oracles $\mathcal{O}_{rk}$, $\mathcal{O}_{re}$ and $\mathcal{O}_{dec}$,[6] the value of the bit b remains hidden to $\mathcal{A}$. This is so because the verification key C_0^* and the signature C_6^* (as well as ID^* and w^*) are independent of b, C_2^*, C_3^*, C_4^*, C_5^* can be regarded as the ciphertext of a 3-level BBG HIBE scheme with CPA security, and the output of *PRF* randomly hides the value of the bit b (assuming *PRF* is picked up from the pseudorandom function family).

We further show that $\mathcal{A}$ cannot obtain additional advantage in guessing the value of the bit b with the help of oracles $\mathcal{O}_{re}$ and $\mathcal{O}_{dec}$. On one hand, if $\mathcal{A}$ submits a query $((ID^*, w^*), C_0^*, C_1', C_2', C_3', C_4', C_5', C_6')$ to $\mathcal{O}_{re}$ and $\mathcal{O}_{dec}$ such that $C_6^* \neq C_6'$, $C_1^* \neq C_1'$, $C_2^* \neq C_2'$, $C_3^* \neq C_3'$, $C_4^* \neq C_4'$ and $C_5^* \neq C_5'$, then the oracles will respond $\perp$ as $\mathcal{A}$ cannot output a new and valid signature C_6' with respect to the same verification key C_0^* (assuming the underlying OTS scheme is strongly existential unforgeable). On the other hand, if $C_0^* \neq C_0'$, then the corresponding re-encryption and decryption will not help $\mathcal{A}$ in guessing the value of the bit b as the ciphertext is generated under a different 3-level identity vector $I = (ID^*, w^*, C_0')$. From the challenge ciphertext, we can see that $\mathcal{A}$ can also submit a tuple $((ID^*, w^*), C_0^*, C_1^*, C_2', C_3^*, C_4^*, C_5^*, C_6^*)$ to the oracles (where $C_2^* \neq C_2'$). If the challenger can verify whether the querying ciphertext is either an invalid one or a derivative of the challenge ciphertext, then $\mathcal{A}$ still cannot obtain any additional advantage in winning the game. This verification depends on the capability of the recovery of C_2^*. This is so because if C_2' is not derived from C_2^* (i.e. C_2^* cannot be deduced from C_2'), and meanwhile $C_2' \neq C_2^*$, then it can be convinced that C_2' is mutated from C_2^* by $\mathcal{A}$. In addition, the first $\lambda_1 - \lambda$ bits of C_1^* can be used to verify the validity of C_2' as well.

We present our formal security proof as follows.

Theorem 2. *Suppose the decisional* 3*-wBDHI assumption holds, the underlying OTS scheme is strongly existential unforgeable (sUF) and PRF is a pseudorandom function family, our BiMH-IBCPRE scheme is IND-sCon-sID-CCA secure in the standard model.*

Please refer to Appendix A for the proof of Theorem 2.

By Theorem 1 and Theorem 2, we have the following corollary.

Corollary 1. *Suppose our BiMH-IBCPRE scheme is IND-sCon-sID-CCA secure in the standard model, it also achieves collusion resistance in the standard model.*

[6] Here the trivial oracle queries are those forbidden queries defined in Definition 2.

Remarks. In the IND-sCon-sID-CCA game defined in Definition 2, we can see that the adversary $\mathcal{A}$ can achieve a re-encryption key from a challenge identity ID_i^* to a corrupted identity ID' under w, where w is not the challenge condition. Despite $\mathcal{A}$ can retrieve the partial re-encryption key $prk_{w,i^*} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6)$ from the above re-encryption key with knowledge of $sk_{ID'}$, it is still unable to recover the entire private key of identity ID_i^* as the random exponents ρ_1, ρ_2 are unknown.

6. An efficient extension

In this section we extend our basic BiMH-IBCPRE scheme to support multiple conditions. In the multiple conditions setting, the proxy is allowed to transform a ciphertext under ID_A tagged with $W = \{w_z \mid 1 \le z \le n, w_z \in \mathbb{Z}_q^*\}$ to another ciphertext under ID_B labeled with the same condition set, *if and only if* it is given a re-encryption key from ID_A to ID_B under W. The extension of our basic scheme is as follows.

1. $Setup(1^\lambda, n)$. Same as that of the basic scheme except that choose a TCR hash function $H_0 : \{0,1\}^* \to \mathbb{Z}_q^*$, choose $h_4, ..., h_{n+2} \in_R \mathbb{G}_1^{n-1}$, and add $H_0, h_4, ..., h_{n+2}$ to mpk.
2. $KeyGen(msk, ID)$. Given the master secret key msk and an identity ID (i.e. an identity $I = (ID)$), choose $r \in_R \mathbb{Z}_q^*$, and output
$$sk_{ID} = \left(g_2^\alpha \cdot \left(h_1^{ID} \cdot g_3\right)^r, g^r, h_2^r, h_3^r, ..., h_{n+2}^r\right) \in \mathbb{G}_1^{n+3}.$$
Given $sk_{ID} = (a_0, a_1, b_2, b_3, ..., b_{n+2})$, $I = (ID, w_1)$, choose a $t \in_R \mathbb{Z}_q^*$, the private key $sk_{(ID,w_1)}$ can be derived from sk_{ID} as
$$\begin{aligned} sk_{(ID,w_1)} &= \left(a_0 \cdot b_2^{w_1} \cdot \left(h_1^{ID} \cdot h_2^{w_1} \cdot g_3\right)^t, a_1 \cdot g^t, b_3 \cdot h_3^t, ..., b_{n+2} \cdot h_{n+2}^t\right) \\ &= \left(g_2^\alpha \cdot \left(h_1^{ID} \cdot h_2^{w_1} \cdot g_3\right)^{r'}, g^{r'}, h_3^{r'}, ..., h_{n+2}^{r'}\right), \end{aligned}$$
where $r' = r + t$.
3. $Enc(ID_i, W, m)$. Same as that of the basic scheme except that $C_4 = (h_1^{ID_i} \cdot h_2^{w_1} \cdots h_{n+1}^{w_n} \cdot h_{n+2}^{K_v} \cdot g_3)^s$ and $C_5 = (f_1^{H_0(W)} \cdot f_2)^s$, and output the ciphertext $C_{(ID_i,W)} = ((ID_i, W), C_0, C_1, C_2, C_3, C_4, C_5, C_6)$.
4. Re-Encryption Key Generation Protocol.
   (a) $PReKeyGen(sk_{ID_j}, W)$. ID_j first generates $sk_{(ID_j,W)} = (a_{0j}, a_{1j}, b_{n+2j})$ (under $I = (ID_j, W)$), chooses $\rho_1, \rho_2 \in_R \mathbb{Z}_q^*$, computes $\beta_1 = {a_{0j}}^{-1} \cdot (f_1^{H_0(W)} \cdot f_2)^{\rho_1}$, $\beta_2 = g^{\rho_1}$, $\beta_3 = {a_{1j}}^{-1} \cdot g^{\rho_2}$, $\beta_4 = b_{n+2j}^{-1} \cdot h_{n+2}^{\rho_2}$, $\beta_5 = (h_2^{w_1} \cdots h_{n+1}^{w_n} \cdot g_3)^{\rho_2}$, $\beta_6 = h_1^{\rho_2}$, and sends the partial re-encryption key $prk_{(W,j)} = (\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6)$ to ID_i.
   (b) $ReKeyGen(sk_{ID_i}, prk_{(W,j)}, W)$. ID_i first generates $sk_{(ID_i,W)} = (a_{0i}, a_{1i}, b_{n+2i})$ (under $I = (ID_i, W)$), chooses $\rho_3, \rho_4 \in_R \mathbb{Z}_q^*$, computes $rk_1 = a_{0i} \cdot \beta_1 \cdot (f_1^{H_0(W)} \cdot f_2)^{\rho_3}$, $rk_2 = g^{\rho_3} \cdot \beta_2$, $rk_3 = a_{1i} \cdot \beta_3 \cdot g^{\rho_4}$, $rk_4 = b_{n+2i} \cdot \beta_4 \cdot h_{n+2}^{\rho_4}$, $rk_5 = (h_2^{w_1} \cdots h_{n+1}^{w_n} \cdot g_3)^{\rho_4} \cdot \beta_5$, $rk_6 = h_1^{\rho_4} \cdot \beta_6$ and outputs $rk_{i \to j|W} = (rk_1, rk_2, rk_3, rk_4, rk_5, rk_6)$.
   (c) The proxy can derive a new re-encryption key $rk_{j \to i|W}$ from $rk_{i \to j|W}$ by running $rk_{j \to i|W} \leftarrow ReKeyBiGen(rk_{i \to j|W})$.
5. $ReEnc(rk_{i \to j|W}, C_{(ID_i,W)})$. Same as that of the basic scheme except that Eq. (1) is modified as:
$$\begin{aligned} &Verify\left(C_0, C_6, (C_1, C_3, C_4, C_5)\right) \stackrel{?}{=} 1, \qquad e\left(C_3, f_1^{H_0(W)} \cdot f_2\right) \stackrel{?}{=} e(g, C_5), \\ &e\left(C_3, h_1^{ID_i} \cdot h_2^{w_1} \cdots h_{n+1}^{w_n} \cdot h_{n+2}^{K_v} \cdot g_3\right) \stackrel{?}{=} e(g, C_4) \end{aligned} \tag{2}$$
After re-encryption, the proxy outputs $C_{(ID_j,W)} = ((ID_i, W), C_0, C_1, C_2^{(l)}, C_3, C_4, C_5, C_6)$ as the re-encrypted ciphertext.
6. $Dec(sk_{ID_i}, C_{(ID_i,W)})$. With knowledge of sk_{ID_i} (under $I = (ID_i)$), user ID_i can generate the private key $sk_{(ID_i,W)} = (a_{0i}, a_{1i}, b_{n+2i})$ (under $I = (ID_i, W)$), and then recover the plaintext as follows.
   (a) Verify Eq. (2). If the equation does not hold, output $\perp$. Otherwise, proceed.
   (b) Compute $\sigma = C_2^{(l)} \cdot \frac{e(a_{1i}, C_4)}{e(a_{0i} \cdot b_{n+2i}^{C_0}, C_3)}$, and verify $[PRF_\sigma(C_3)]^{\lambda_1 - \lambda} \stackrel{?}{=} [C_1]^{\lambda_1 - \lambda}$. If the equation holds, output $m = [C_1]_\lambda \oplus [PRF_\sigma(C_3)]_\lambda$. Otherwise, output $\perp$.

From the above extension, it is interesting to see that the size of ciphertext and the number of bilinear pairings used in decryption is independent of the number of conditions. Thus, our extension does not incur a great deal of cost with respect to computation and communication. Similar to the basic scheme, the IND-sCon-sID-CCA security of the extended scheme can be proved in the standard model under the decisional $(n+2)$-wBDHI assumption. The proof of the basic scheme presented in Appendix A can be easily adapted to that of the extension, we hence omit the details.

7. Comparison

In this section we compare our basic scheme with [7,9,14,27] in terms of properties and efficiency. Table 2 shows the comparison of computation cost, and Table 3 shows the comparison of communication complexity.

Table 2
Computation cost comparison.

Sch.	Computation cost				
	Enc	*ReEnc*	*Dec* (1st level)	*Dec* (2nd level)	*ReKey*
[7]	$5t_e + t_p + t_s$	$t_e + t_v + 4t_p$	$t_e + t_v + 5t_p$	$t_e + t_v + 5t_p$	–
[14]	$2t_e + t_p$	t_p	t_p	$2t_p$	$2t_e + t_p$
[9]	$4t_e + t_p + t_s$	$2t_e + t_v$	$2t_e + 3t_p + t_v$	$2t_e + 10t_p + 2t_v$	$4t_e + t_p + t_s$
[27]	$21t_e + t_p + t_s + t_{SKE}$	$29t_e + 18t_p + t_s + t_v + t_{SKE}$	$8t_e + 17t_p + t_v + t_{SKE}$	$18t_e + 34t_p + 3t_v + 3t_{SKE}$	$38t_e + t_p + t_s + t_{SKE}$
Ours	$8t_e + t_p + t_s + t_{PRF}$	$6t_e + 7t_p + t_v$	$5t_e + 6t_p + t_v + t_{PRF}$	$5t_e + 6t_p + t_v + t_{PRF}$	$16t_e$

Table 3
Communication complexity comparison.

Sch.	Ciphertext and re-encryption key length		
	Original ciphertext	Re-encrypted ciphertext	*ReKey*
[7]	$3\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + \|svk\| + \|\sigma\|$	$3\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + \|svk\| + \|\sigma\|$	$\|\mathbb{Z}_q\|$
[14]	$\|\mathbb{G}_1\| + \|\mathbb{G}_T\|$	$2\|\mathbb{G}_1\| + 2\|\mathbb{G}_T\|$	$2\|\mathbb{G}_1\| + \|\mathbb{G}_T\|$
[9]	$3\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + \|svk\| + \|\sigma\|$	$9\|\mathbb{G}_1\| + 2\|\mathbb{G}_T\| + 2\|svk\| + 2\|\sigma\|$	$5\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + \|svk\| + \|\sigma\|$
[27]	$11\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + 3\|\mathbb{Z}_q\| + \|svk\| + \|\sigma\| + \|SKE\|$	$22\|\mathbb{G}_1\| + 3\|\mathbb{G}_T\| + 6\|\mathbb{Z}_q\| + 3\|svk\| + 3\|\sigma\| + 3\|SKE\|$	$21\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + 5\|\mathbb{Z}_q\| + \|svk\| + \|\sigma\| + \|SKE\|$
Ours	$3\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + \|svk\| + \|\sigma\| + \lambda_1$	$3\|\mathbb{G}_1\| + \|\mathbb{G}_T\| + \|svk\| + \|\sigma\| + \lambda_1$	$6\|\mathbb{G}_1\|$

We first define the notations and parameters used in the tables. Let $|\mathbb{G}_1|$ and $|\mathbb{G}_T|$ denote the bit-length of an element in groups $\mathbb{G}_1$ and $\mathbb{G}_T$, $|svk|$ and $|\sigma|$ denote the bit-length of the verification key and signature of OTS, $|SKE|$ denote the bit-length of the ciphertext of a one-time symmetric key encryption, λ_1 denote the security parameter, $t_p, t_e, t_v, t_s, t_{PRF}, t_{SKE}$ denote the computation cost of a bilinear pairing, an exponentiation, one verification and one signature of an OTS scheme, a pseudorandom function and a one-time symmetric key encryption, respectively.

We now analyze the property comparison shown in Table 1. Generally, our scheme with constant-size ciphertext is CCA secure in the standard model under the decisional 3-wBDHI assumption; meanwhile, it is the only scheme supporting conditional re-encryption. In addition, only [27] and our scheme achieve collusion resistance. Specifically, when compared with [14] (the scheme built in the random oracle model) which is only CPA secure under the decisional bilinear Diffie–Hellman (DBDH) assumption [33], our scheme achieves stronger security and enjoys more properties. [9] is RCCA secure under the DBDH assumption but its ciphertext's size grows linearly in the number of re-encryption hops; nonetheless, our scheme is secure against CCA, and the size of ciphertext remains constant. Despite [7] (the scheme constructed in the standard model) is proved CCA secure under the modified DBDH (mDBDH) assumption [26], it cannot support identity-based re-encryption and conditional re-encryption properties. Compared to [27], our scheme enjoys additional properties, including conditional re-encryption and constant size ciphertext without losing CCA security. Moreover, [27][7] requires that the underlying HIBE scheme must be fully CPA (i.e. IND-aID-CPA) secure. Nevertheless, our scheme is not limited to such a strong restriction but only requires the underlying HIBE scheme to be IND-sID-CPA secure. In conclusion, our scheme enables system users to implement more fine-grained delegation, and meanwhile, lessens the cost of storage for cloud server as well as the communication bandwidth without degrading security level. However, the problem of proposing a fully CCA-secure MH-IBCPRE in the standard model achieving all the properties of our scheme remains open.

Without loss of fairness, we assume all the schemes listed in tables share the same number of re-encryption hops N, and for simplicity, we further set $N = 1$; meanwhile, the identity associated with ciphertext can be seen as public information in [9,27] and our scheme. Note that in our scheme (ID_i, w) is regarded as a single identity I. From Table 2, we see that [14] is the most efficient scheme, while [27] suffers from the largest number of pairings. Compared with [7,9], our scheme increases $1t_p$ and reduces $4t_p$ in *Dec* (2nd level), respectively. Despite the overall performance of [14] is better than ours, our scheme needs no pairings in *ReKey*. Note that only [7] and our scheme require no pairings in *ReKey*. As opposed to [27], our scheme significantly reduces the number of pairings in *ReEnc* and *Dec*. Thus, despite the performance of our scheme is not better than that of [14] which depends on random oracles, our scheme outperforms [27] which is the only MH-IBPRE with CCA security in the standard model; meanwhile, the cost of pairings in the decryption of our scheme is less than that of [9]. It is worth mentioning that the decryption cost of [9,14,27] is linear in N; on the contrary, ours remains constant.

Table 3 generally shows that [14] has the smallest number of groups of elements for the ciphertexts and [7] only requires one group of element in *ReKey*, while [27] suffers from the largest number of groups of elements in each metric. Despite the overall communication cost of [14] is less than ours, our scheme reduces two elements in $\mathbb{G}_T$. When compared with [7], one additional group of elements is needed in our ciphertext; in *ReKey*, we need 6 elements in $\mathbb{G}_1$, while [7] only requires one element in $\mathbb{Z}_q$. [9] enjoys less groups of elements in original ciphertext, but our scheme reduces 6 elements in $\mathbb{G}_1$ and 1 element in $\mathbb{G}_T, svk, \sigma$, respectively, in terms of re-encrypted ciphertext. Besides, our re-encryption key only includes one group of elements when that of [9] requires 4. It is not difficult to see that our scheme outperforms [27], especially in *ReKey*.

[7] In [27], Shao and Cao first proposed a generic construction for MH-IBPRE that requires an HIBE as primitive, and next presented a concrete scheme. In Table 2, we compare our scheme with their concrete MH-IBPRE, which is secure against CCA under the decisional linear [34] and DBDH assumptions.

In conclusion, our scheme has better overall performance in communication when compared with [9,27]; meanwhile, it enjoys more properties than [7,14] without significantly increasing the communication cost. Note our scheme will achieve better efficiency when N increases as our ciphertext size remains constant but [14,9,27] do not.

8. Applications of BiMH-IBCPRE

Groups data sharing. In a company, working groups of different departments might cooperate with each other so as to accelerate the process of some business project. For example, a project $P1$ is assigned to group A, group B and group C simultaneously. Suppose the project data is stored in a cloud storage system. To enable each worker in the groups to share the data of $P1$ without suffering from large decryption and encryption complexity, we can employ a BiMH-IBCPRE in the system.

The managers of Group A and group B can first generate a re-encryption key $rk_{A\to B|P1}$ from A to B under a condition $P1$. Likewise, the managers of group B and group C can generate another re-encryption key $rk_{B\to C|P1}$ from B to C under $P1$. The managers then upload the re-encryption keys to the cloud (i.e. the proxy). Due to the derivation property of re-encryption key, the proxy can deduce new re-encryption keys $rk_{B\to A|P1}$ and $rk_{C\to B|P1}$ from the existing ones. With these re-encryption keys the proxy can transform the ciphertexts of the data to each group. For instance, by using $rk_{A\to B|P1}$ the proxy can convert a ciphertext $Enc(ID_A, P1, m_1)$ of a data m_1, stored in the cloud by group A, to $Enc(ID_B, P1, m_1)$, which can be only decrypted by group B. Similarly, the ciphertexts encrypted under ID_B and $P1$ can be converted to the ones intended for group A.

Even if one of the groups, say group C, might simultaneously deal with another project, say $P2$, the fine-grained control on re-encryption disallows group B (resp. A) to gain access to any data associated with a condition $P2$. This is so because the re-encryption key tagged with a condition $P1$ only allows the proxy to convert the ciphertexts associated with the same condition. Specifically, the re-encryption key $rk_{B\to C|P1}$ and its derivation $rk_{C\to B|P1}$ only allow the transformation of ciphertexts between B and C under project $P1$.

Furthermore, since BiMH-IBCPRE supports multiple conditions, the working groups can make a more fine-grained control on data sharing. For example, group B and group C may choose to bidirectionally share their data tagged with a keyword set $W = (\text{“}P1\text{”}, \text{“stage 2”}, \text{“meeting in company”}, \text{“03/04/2012”})$. Here as long as the corresponding re-encryption key $rk_{B\to C|W}$ (resp. $rk_{C\to B|W}$) is given to the proxy, the ciphertexts of the data matching the keyword set can be converted between B and C.

Multi-devices data sharing. With popularization of cloud storage, many Internet users prefer to backup their data, which are stored in some personal portable devices (e.g., Ipad and mp3), to the cloud. How to efficiently and securely share the data among such devices becomes an interesting issue. Suppose an Internet user, say Alice, has three portable electronic devices, a PDA A_1, an Ipad A_2 and a laptop A_3. She might choose to share some data, e.g., a photo m tagged with a description S, among her devices by using the technology of BiMH-IBCPRE as follows.

Without loss of confidentiality, Alice prefers to encrypt the photo as $Enc(ID_{A_1}, S, m)$ before uploading to the cloud via A_1 (i.e. Alice's PDA). Obtaining the fully control on her devices, Alice can generate the re-encryption keys $rk_{A_1\to A_2|S}$ and $rk_{A_2\to A_3|S}$ for the cloud server. The server, acting as a proxy, then re-encrypts the ciphertext of m under ID_{A_1} and S to $Enc(ID_{A_2}, S, m)$, which can be only decrypted in A_2. Furthermore, the server can re-encrypt the resulting ciphertext to $Enc(ID_{A_3}, S, m)$. When accessing the photo from A_2, Alice first logs in the device, and next decrypts $Enc(ID_{A_2}, S, m)$ after downloading from the cloud. Similarly, Alice can read the photo in A_3.

In this data sharing mechanism, even if one of the devices (e.g., A_1) is crashed down, the ciphertext of the photo stored in the cloud can still be accessed via the remaining devices, A_2 and A_3. If the device A_1 is lost or stolen by an adversary, Alice can request the proxy to delete the corresponding re-encryption key related to A_1, e.g., $rk_{A_1\to A_2|S}$ and $rk_{A_2\to A_1|S}$, so as to prevent the unauthorized adversary from accessing the data of the remaining devices via A_1. Although it might be possible for the adversary to obtain the re-encryption key $rk_{A_1\to A_2|S}$ from the proxy by intrusion, the adversary cannot compromise the entire private key of A_2 due to the collusion resistance property. In addition, even in the worst case, that is, the adversary obtains $rk_{A_1\to A_2|S}$ and further converts the ciphertexts of A_2 (stored in the cloud) to the ones which can be decrypted in A_1, the system still guarantees that only the data associated with S other than all the data of A_2 will leak to the adversary.

9. Conclusion

In this paper we proposed the first BiMH-IBPRE scheme that achieves conditional re-encryption with constant-size ciphertext. Moreover, we proved our scheme to be IND-sCon-sID-CCA secure and collusion resistant in the standard model. When compared with the existing MH-IBPRE schemes, our scheme achieves better performance in terms of communication complexity, and is able to provide fine-grained decryption rights delegation for system users.

This paper also motivates some interesting open problems, for example, how to construct a CCA-secure BiMH-IBCPRE system in the adaptive condition and adaptive identity model, i.e. achieving IND-aCon-aID-CCA security.

Appendix A. Proof of Theorem 2

Proof. Suppose an adversary $\mathcal{A}$ can break the IND-sCon-sID-CCA security of our scheme. We then construct a reduction algorithm $\mathcal{C}_1$ to solve the decisional 3-wBDHI* problem using $\mathcal{A}$. Before proceeding, we first consider the sUF of the underlying one-time signature scheme. Suppose the challenge ciphertext is $((ID^*, w^*), C_0^*, C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$. Let $event_1$ be the case that $\mathcal{A}$ issues a decryption query $((ID^*, w^*), C_0^*, C_1', C_2'^{(l)}, C_3', C_4', C_5', C_6')$, where $Verify(C_0^*, C_6', (C_1', C_3', C_4', C_5')) = 1$. Let $event_2$ be the case that $\mathcal{A}$ issues a re-encryption query $((ID^*, w^*), C_0^*, C_1', C_2'^{(l)}, C_3', C_4', C_5', C_6')$, where $Verify(C_0^*, C_6', (C_1', C_3', C_4', C_5')) = 1$. If one of the above cases occurs, we can construct an algorithm $\mathcal{C}_2$ to break the sUF of the underlying one-time signature scheme. In the simulation $\mathcal{C}_1$ will output a random bit and abort the simulation when either $event_1$ or $event_2$ occurs.

1. **Init.** $\mathcal{A}$ outputs the challenge identity ID^* and the challenge condition w^* (i.e. $I^* = (I_1^*, I_2^*) = (ID^*, w^*)$) to $\mathcal{C}_1$.
2. **Setup.** $\mathcal{C}_1$ takes in $(q, g, \mathbb{G}_1, \mathbb{G}_T, e) \leftarrow BSetup(1^\lambda)$ and a problem instance (g, h, y_1, y_2, y_3, T) that is either sampled from $P_{3\text{-wBDHI}^*}$ (where $T = e(g,h)^{\alpha^4}$) or from $R_{3\text{-wBDHI}^*}$ (where $T \in_R \mathbb{G}_T$), where $y_1 = g^\alpha$, $y_2 = g^{\alpha^2}$ and $y_3 = g^{\alpha^3}$. To generate mpk, $\mathcal{C}_1$ prepares a pseudorandom function $PRF : \mathbb{G}_T \times \mathbb{G}_1 \to \{0,1\}^{\lambda_1}$, runs $(K_v^*, K_s^*) \leftarrow Sign.KeyGen(1^\lambda)$, and then sets the 3-level identity $I^* = (I_1^*, I_2^*, I_3^*) = (ID^*, w^*, K_v^*)$ as a challenge identity for a 3-level HIBE game. $\mathcal{C}$ further chooses γ, γ_1, γ_2, γ_3, γ_4, γ_5, $\xi \in_R \mathbb{Z}_q^*$ and sets $f_1 = g^{\gamma_4}$, $f_2 = g^{\gamma_5}$, $g_1 = y_1 = g^\alpha$, $g_2 = y_3 \cdot g^\gamma = g^{\gamma+\alpha^3}$, $h_1 = \frac{g^{\gamma_1}}{y_3}$, $h_2 = \frac{g^{\gamma_2}}{y_2}$, $h_3 = \frac{g^{\gamma_3}}{y_1}$ and $g_3 = g^\xi \cdot \prod_{i=1}^3 y_{4-i}^{I_i^*}$. Finally, $\mathcal{C}_1$ outputs $mpk = (q, g, \mathbb{G}_1, \mathbb{G}_T, e, f_1, f_2, g_1, g_2, g_3, h_1, h_2, h_3, PRF, (Sign.KeyGen, Sign, Verify))$. The distribution of mpk is identical to that of the real scheme in $\mathcal{A}$'s view. Note that msk is $g_2^\alpha = g^{\alpha\cdot(\alpha^3+\gamma)} = y_4 \cdot y_1^\gamma$ which cannot be computed by $\mathcal{C}_1$, where $y_4 = g^{\alpha^4}$. Besides, $\mathcal{C}_1$ maintains the following lists which are initially empty.
   (a) DCT: records the tuples $(w, K_v, ID_{root}, ..., ID_i, ..., tag_1, tag_2)$,[8] which are the re-encryption chains under w and K_v starting from ID_{root} and including ID_i, where tag_1, tag_2 denote that the chain is either uncorrupted or corrupted. $\mathcal{C}_1$ uses it to record the transformation.
   (b) SKT: records the tuples $(ID_i, w, K_v, a_{0i}, a_{1i}, b_{2i}, b_{3i}, tag)$, which are the private keys' information of uncorrupted or corrupted users, where $sk_{ID_i} = (a_{0i}, a_{1i}, b_{2i}, b_{3i})$, tag denotes whether it can be achieved by $\mathcal{A}$ or not.
   (c) RKT: records the tuples $(ID_i, ID_j, w, rk_{i\to j|w}, prk_{(w,j)}, tag_1, tag_2, tag_3)$, which are the results of the queries to $\mathcal{O}_{rk}(ID_i, ID_j, w)$, where tag_1, tag_2, tag_3 denote that $rk_{i\to j|w}$ is randomly chosen, generated in $\mathcal{O}_{re}$ or in $\mathcal{O}_{rk}$.
   (d) RET: records the tuples $(ID_i, ID_j, w, C_{(ID_j,w)}, tag_1, tag_2, tag_3)$, which are the results of the queries to $\mathcal{O}_{re}(ID_i, ID_j, w, C_{(ID_i,w)})$, where tag_1, tag_2, tag_3 denote that the ciphertext is re-encrypted using a real re-encryption key, a random re-encryption key or is generated without any re-encryption key.
3. **Phase 1.** $\mathcal{A}$ issues a series of queries to which $\mathcal{C}_1$ responds as follows.
   (a) $\mathcal{O}_{sk}(ID)$: if $ID = ID^*$ or, ID is in an uncorrupted delegation chain including ID^*,[9] $\mathcal{C}_1$ outputs a random bit in $\{0,1\}$ and aborts. Otherwise,
   - If $(ID, \perp, \perp, a_{0i}, a_{1i}, b_{2i}, b_{3i}, 1) \in SKT$: $\mathcal{C}_1$ outputs sk_{ID}.
   - Else, $\mathcal{C}_1$ chooses a $r_{ID} \in_R \mathbb{Z}_q^*$, sets $sk_{ID} = (a_0, a_1, b_2, b_3) = (y_1^\gamma \cdot (g^\xi \cdot g^{ID\cdot\gamma_1})^{r_{ID}} \cdot (y_1^\xi \cdot y_1^{ID\cdot\gamma_1} \cdot y_2^{K_v^*} \cdot y_3^{w^*})^{\frac{1}{ID-ID^*}} \cdot (y_1^{K_v^*} \cdot y_2^{w^*} \cdot y_3^{(ID^*-ID)})^{r_{ID}}, g^{r_{ID}} \cdot y_1^{\frac{1}{ID-ID^*}}, h_2^{r_{ID}} \cdot (\frac{y_1^{\gamma_2}}{y_3})^{\frac{1}{ID-ID^*}}, h_3^{r_{ID}} \cdot (\frac{y_1^{\gamma_3}}{y_2})^{\frac{1}{ID-ID^*}})$. Let $r = r_{ID} + \frac{\alpha}{ID-ID^*}$, one can verify

$$\begin{aligned}
a_0 &= y_1^\gamma \cdot \left(g^\xi \cdot g^{ID\cdot\gamma_1}\right)^{r_{ID}} \cdot \left(y_1^\xi \cdot y_1^{ID\cdot\gamma_1} \cdot y_2^{K_v^*} \cdot y_3^{w^*}\right)^{\frac{1}{ID-ID^*}} \cdot \left(y_1^{K_v^*} \cdot y_2^{w^*} \cdot y_3^{(ID^*-ID)}\right)^{r_{ID}} \\
&= y_1^\gamma \cdot \left(g^{\xi+ID\cdot\gamma_1}\right)^r \cdot y_1^{K_v^*\cdot r} \cdot y_2^{w^*\cdot r} \cdot y_3^{r_{ID}\cdot(ID^*-ID)} \\
&= \left(y_4 \cdot y_1^\gamma\right) \cdot \left(g^{\xi+ID\cdot\gamma_1} \cdot y_1^{K_v^*} \cdot y_2^{w^*}\right)^r \cdot \left(y_3^{r_{ID}\cdot(ID^*-ID)}/y_4\right) \\
&= \left(y_4 \cdot y_1^\gamma\right) \cdot \left(g^{\xi+ID\cdot\gamma_1} \cdot y_1^{K_v^*} \cdot y_2^{w^*}\right)^r \cdot y_3^{r\cdot(ID^*-ID)} \\
&= g_2^\alpha \cdot \left(g^\xi \cdot y_1^{K_v^*} \cdot y_2^{w^*} \cdot y_3^{ID^*} \cdot g^{ID\cdot\gamma_1} \cdot y_3^{-ID}\right)^r \\
&= g_2^\alpha \cdot \left(h_1^{ID} \cdot g_3\right)^r,
\end{aligned}$$

   and $g^{r_{ID}} \cdot y_1^{\frac{1}{ID-ID^*}} = g^{r_{ID}} \cdot g^{\frac{\alpha}{ID-ID^*}} = g^r$, $h_2^{r_{ID}} \cdot (\frac{y_1^{\gamma_2}}{y_3})^{\frac{1}{ID-ID^*}} = h_2^{r_{ID}} \cdot (\frac{g^{\gamma_2}}{y_2})^{\frac{\alpha}{ID-ID^*}} = h_2^r$, $h_3^{r_{ID}} \cdot (\frac{y_1^{\gamma_3}}{y_2})^{\frac{1}{ID-ID^*}} = h_3^{r_{ID}} \cdot (\frac{g^{\gamma_3}}{y_1})^{\frac{\alpha}{ID-ID^*}} = h_3^r$. $\mathcal{C}_1$ then returns sk_{ID} to $\mathcal{A}$ and adds $(ID, \perp, \perp, a_{0i}, a_{1i}, b_{2i}, b_{3i}, 1) \in SKT$.

[8] $(w, K_v, ID_{root}, \perp, *, *)$ indicates that there is an original ciphertext generated under ID_{root} and w, K_v.

[9] It has three cases: $(w^*, *, ID, ..., 1, 0)$, $(w^*, *, *, ..., ID, ..., 1, 0)$, or $(w^*, *, *, ..., ID, 1, 0)$. If the identity is in a corrupted delegation chain, we have $(w^*, *, ID, ..., 0, 1)$, $(w^*, *, *, ..., ID, ..., 0, 1)$, or $(w^*, *, *, ..., ID, 0, 1)$ in DCT, where $*$ is the wildcard. Hereafter, by saying an identity is in a delegation chain we mean that there is one of the above cases in DCT.

(b) $\mathcal{O}_{rk}(ID_i, ID_j, w)$: If $\mathcal{A}$ issues a query (ID_i, ID_j, w) to $\mathcal{O}_{rk}$ to receive a re-encryption key $rk_{i\to j|w}$, then it can achieve a new re-encryption key $rk_{j\to i|w}$ via algorithm *ReKeyBiGen*. Similarly, when $rk_{i\to j|w}$ is generated by $\mathcal{C}$, $\mathcal{C}$ can answer the query (ID_j, ID_i, w) via algorithm *ReKeyBiGen*. In this simulation, set $k \in \{i, j\}$. If $(ID_{i+j-k}, ID_k, w, rk_{i+j-k\to k|w}, prk_{(w,k)}, *, 0, 1) \in RKT$, $\mathcal{C}_1$ returns $rk_{i+j-k\to k|w}$ to $\mathcal{A}$. Otherwise,

- If ID^* will be in a corrupted delegation chain under w^* after generating the re-encryption key, then $\mathcal{C}_1$ aborts. Specifically, the following queries are forbidden to issue:
  i. $w^* = w$, $ID^* = ID_{i+j-k}$, and ID_k is either a corrupted identity, i.e. $(ID_k, \perp, \perp, a_{0k}, a_{1k}, b_{2k}, b_{3k}, 1) \in SKT$, or in a corrupted delegation chain under w.
  ii. $w^* = w$, ID_{i+j-k} is in an uncorrupted delegation chain, and ID_k is either a corrupted identity, or in a corrupted delegation chain under w.
- If $w^* = w$, $ID^* = ID_{i+j-k}$, and ID_k is either an uncorrupted identity or in an uncorrupted delegation chain under w, $\mathcal{C}_1$ checks whether $(ID_{i+j-k}, ID_k, w, rk_{i+j-k\to k|w}, prk_{(w,k)}, 1, 1, 0) \in RKT$. If yes, $\mathcal{C}_1$ returns $rk_{i+j-k\to k|w}$ and resets $tag_2 = 0$, $tag_3 = 1$. Else, $\mathcal{C}_1$ checks whether sk_{ID_k} (i.e. the tuple $(ID_k, \perp, \perp, a_{0k}, a_{1k}, b_{2k}, b_{3k}, *)$) is in *SKT*. If not, $\mathcal{C}_1$ constructs $sk_{ID_k} = (a_{0k}, a_{1k}, b_{2k}, b_{3k}) = (g_2^{\alpha} \cdot (h_1^{ID_k} \cdot g_3)^r, g^r, h_2^r, h_3^r)$ as step (a), and adds the corresponding tuple to *SKT*. Else, $\mathcal{C}_1$ derives $sk_{(ID_k,w^*)}$ from sk_{ID_k} as $sk_{(ID_k,w^*)} = (a_{0k} \cdot b_{2k}^{w^*} \cdot (h_1^{ID_k} \cdot h_2^{w^*} \cdot g_3)^t, a_{1k} \cdot g^t, b_{3k} \cdot h_3^t) = (g_2^{\alpha} \cdot (h_1^{ID_k} \cdot h_2^{w^*} \cdot g_3)^{r'}, g^{r'}, h_3^{r'}) = (a'_{0k}, a'_{1k}, b'_{3k})$, where $t \in_R \mathbb{Z}_q^*$, $r' = r + t$. $\mathcal{C}_1$ further checks whether $(ID^*, w^*, \perp, a_{0*}, a_{1*}, \perp, b_{3*}, 0) \in SKT$. If not, $\mathcal{C}_1$ chooses a_{0*}, a_{1*}, $b_{3*} \in_R \mathbb{G}_1$ and adds the corresponding tuple to *SKT*. Otherwise, $\mathcal{C}_1$ constructs $prk_{(w^*,k)}$ as: $\beta_1 = a'^{-1}_{0k} \cdot (f_1^{w^*} \cdot f_2)^{\rho_1}$, $\beta_2 = g^{\rho_1}$, $\beta_3 = a'^{-1}_{1k} \cdot g^{\rho_2}$, $\beta_4 = b'^{-1}_{3k} \cdot h_3^{\rho_2}$, $\beta_5 = (h_2^{w^*} \cdot g_3)^{\rho_2}$, $\beta_6 = h_1^{\rho_2}$, and $rk_{i+j-k\to k|w^*}$ as: $rk_1 = a_{0*} \cdot \beta_1 \cdot (f_1^{w^*} \cdot f_2)^{\rho_3}$, $rk_2 = g^{\rho_3} \cdot \beta_2$, $rk_3 = a_{1*} \cdot g^{\rho_4} \cdot \beta_3$, $rk_4 = b_{3*} \cdot \beta_4 \cdot h_3^{\rho_4}$, $rk_5 = (h_2^{w^*} \cdot g_3)^{\rho_4} \cdot \beta_5$, $rk_6 = h_1^{\rho_4} \cdot \beta_6$, where $\rho_1, \rho_2, \rho_3, \rho_4 \in_R \mathbb{Z}_q^*$. Finally, $\mathcal{C}_1$ returns $rk_{i+j-k\to k|w^*}$, updates *DCT* and adds $(ID_{i+j-k}, ID_k, w^*, rk_{i+j-k\to k|w^*}, prk_{(w^*,k)}, 1, 0, 1)$ to *RKT*.
- If $w^* \neq w$ and $ID^* = ID_{i+j-k}$, $\mathcal{C}_1$ checks whether $(ID^*, w, \perp, a_{0*}, a_{1*}, \perp, b_{3*}, 0) \in SKT$. If not, $\mathcal{C}_1$ chooses $r_{ID^*} \in_R \mathbb{Z}_q^*$, constructs $sk_{(ID^*,w)} = (a_{0*}, a_{1*}, b_{3*}) = (y_1^{\gamma} \cdot (g^{\xi+\sum_{i=1}^{2} I_i\cdot\gamma_i} \cdot y_1^{K_v^*})^{r_{ID^*}} \cdot (y_2^{\xi+\sum_{i=1}^{2} I_i\cdot\gamma_i} \cdot y_3^{K_v^*})^{\frac{1}{w-w^*}} \cdot y_2^{r_{ID^*}\cdot(w^*-w)}$, $g^{r_w} \cdot y_2^{\frac{1}{w-w^*}}$, $h_3^{r_w} \cdot (\frac{y_2^{\gamma_3}}{y_3})^{\frac{1}{w-w^*}})$, and adds the corresponding tuple to *SKT*, where $I = (I_1, I_2) = (ID^*, w)$. $\mathcal{C}_1$ then derives $sk_{(ID_k,w)}$ from sk_{ID_k},[10] generates $rk_{i+j-k\to k|w}$ by using $sk_{(ID^*,w)}$ and $sk_{(ID_k,w)}$ as in the real scheme, finally updates *DCT* and adds $(ID_{i+j-k}, ID_k, w, rk_{i+j-k\to k|w}, prk_{(w,k)}, 0, 0, 1)$ to *RKT*.
- Else, $\mathcal{C}_1$ first checks whether $(ID_{i+j-k}, ID_k, w, rk_{i+j-k\to k|w}, prk_{(w,k)}, 0, 1, 0) \in RKT$. If yes, $\mathcal{C}_1$ returns $rk_{i+j-k\to k|w}$ and resets $tag_2 = 0$, $tag_3 = 1$. Else, $\mathcal{C}_1$ checks whether $sk_{ID_{i+j-k}}$ and sk_{ID_k} are in *SKT*. If not, $\mathcal{C}_1$ constructs two private keys as step (a), and adds $sk_{ID_{i+j-k}}$, sk_{ID_k} to *SKT*. Else, $\mathcal{C}_1$ derives $sk_{(ID_{i+j-k},w)}$ and $sk_{(ID_k,w)}$ from $sk_{ID_{i+j-k}}$ and sk_{ID_k}, respectively. $\mathcal{C}_1$ further generates $rk_{i+j-k\to k|w}$ and $prk_{(w,k)}$ from $sk_{(ID_{i+j-k},w)}$ and $sk_{(ID_k,w)}$ as in the real scheme, and returns $rk_{i+j-k\to k|w}$. Finally, $\mathcal{C}_1$ updates *DCT*, adds $(ID_{i+j-k}, ID_k, w, rk_{i+j-k\to k|w}, prk_{(w,k)}, 0, 0, 1)$ to *RKT*.

(c) $\mathcal{O}_{re}(ID_i, ID_j, w, C_{(ID_i,w)})$: $\mathcal{C}_1$ verifies Eq. (1) and the validity of $C_2^{(l)}$.[11] If the verification does not hold, $\mathcal{C}_1$ outputs $\perp$. Otherwise, set $k \in \{i, j\}$ and proceed.

- If the first case of step (b) does not hold: $\mathcal{C}_1$ checks whether $(ID_{i+j-k}, ID_k, w, rk_{i+j-k\to k|w}, prk_{w,k}, *, 0, *) \in RKT$. If yes, $\mathcal{C}_1$ re-encrypts $C_{(ID_{i+j-k},w)}$ by using $rk_{i+j-k\to k|w}$, and adds $(ID_{i+j-k}, ID_k, w, C_{(ID_k,w)}, *, *, 0)$ to *RET*. Else, $\mathcal{C}_1$ constructs the re-encryption key as step (b), re-encrypts the ciphertext as above, updates *DCT* and adds $(ID_{i+j-k}, ID_k, w, rk_{i+j-k\to k|w}, prk_{(w,k)}, *, *, 0)$ and $(ID_{i+j-k}, ID_k, w, C_{(ID_k,w)}, *, *, 0)$ to *RKT* and *RET*, respectively.
- Else, $\mathcal{C}_1$ transforms the ciphertexts under ID^*, w^* and K_v to the ones which can be decrypted by $\mathcal{A}$. Note these re-encryptions can be constructed as long as the ciphertexts taken as input are generated by $\mathcal{A}$. Here $K_v^* \neq K_v$ as K_v^* is chosen by $\mathcal{C}_1$. $\mathcal{C}_1$ checks whether $(ID_k, w^*, K_v, a_{0k}, a_{1k}, \perp, \perp, *) \in SKT$. If not, $\mathcal{C}_1$ constructs sk_{ID_k} as step (a), derives $sk_{(ID_k,w^*,K_v)} = (a_{0k}, a_{1k})$ from sk_{ID_k} and adds the corresponding tuple to *SKT*. Otherwise,
  – If $ID^* = ID_{i+j-k}$,
    i. if $w^* = w$, $ID_{root} = ID_{i+j-k}$, and ID_k is either a corrupted identity or in a corrupted delegation chain, $\mathcal{C}_1$ checks whether $sk_{(ID^*,w^*,K_v)}$ (i.e. the tuple $(ID^*, w^*, K_v, a_{0*}, a_{1*}, \perp, \perp, 0)$) is in *SKT*. If not, $\mathcal{C}_1$ constructs the private key $sk_{(ID^*,w^*,K_v)}$ as $a_{0*} = y_1^{\gamma} \cdot (g^{\xi+\sum_{i=1}^{3} I_i\cdot\gamma_i})^{r_{ID^*}} \cdot (y_3^{\xi+\sum_{i=1}^{3} I_i\cdot\gamma_i})^{\frac{1}{K_v-K_v^*}} \cdot y_1^{r_{ID^*}(K_v^*-K_v)}$, $a_{1*} = y_3^{\frac{1}{K_v-K_v^*}} \cdot g^{r_{ID^*}}$, and then adds the corresponding tuple to *SKT*, where $r_{ID^*} \in_R \mathbb{Z}_q^*$, $I = (I_1, I_2, I_3) = (ID^*, w^*, K_v)$. Let $r = r_{ID^*} + \frac{\alpha^3}{K_v-K_v^*}$, one can verify

$$\begin{aligned} a_{0*} &= y_1^{\gamma} \cdot \left(g^{\xi+\sum_{i=1}^{3} I_i\cdot\gamma_i}\right)^{r_{ID^*}} \cdot \left(y_3^{\xi+\sum_{i=1}^{3} I_i\cdot\gamma_i}\right)^{\frac{1}{K_v-K_v^*}} \cdot y_1^{r_{ID^*}\cdot(K_v^*-K_v)} \\ &= y_1^{\gamma} \cdot \left(g^{\xi+\sum_{i=1}^{3} I_i\cdot\gamma_i}\right)^{r} \cdot y_1^{r_{ID^*}\cdot(K_v^*-K_v)} \end{aligned}$$

[10] If sk_{ID_k} is not in *SKT*, it will be constructed as step (a).

[11] It is not difficult to verify the integrity of $C_2^{(l)}$ as $\mathcal{C}_1$ can construct any private key $sk_{(ID,w,K_v)}$ except for that with $ID^* = ID$, $w^* = w$ and $K_v^* = K_v$. With knowledge of the private keys $\mathcal{C}_1$ can decrypt the ciphertext (issued by $\mathcal{A}$) to verify the integrity of $C_2^{(l)}$ by checking $[C_1]^{\lambda_1-\lambda} \stackrel{?}{=} [PRF_{\sigma}(C_3)]^{\lambda_1-\lambda}$.

$$= g_2^{\alpha} \cdot \left(g^{\xi+\sum_{i=1}^{3} I_i \cdot \gamma_i} \cdot y_1^{K_v^* - K_v}\right)^r$$
$$= g_2^{\alpha} \cdot \left(g^{\xi+\sum_{i=1}^{3} I_i \cdot \gamma_i} \cdot \left(y_2^{w^* - w^*} \cdot y_3^{ID^* - ID^*}\right) \cdot y_1^{K_v^* - K_v}\right)^r$$
$$= g_2^{\alpha} \cdot \left(h_1^{ID^*} \cdot h_2^{w^*} \cdot h_3^{K_v} \cdot g_3\right)^r,$$

and $a_{1*} = g^{r_{ID^*}} \cdot y_3^{\frac{1}{K_v - K_v^*}} = g^{r_{ID^*}} \cdot g^{\frac{\alpha^3}{K_v - K_v^*}} = g^r$. Else, $\mathcal{C}_1$ constructs $C_2^{(2)} = \frac{C_2^{(1)} \cdot e(a_{1*} \cdot (a_{1k})^{-1}, C_4)}{e(a_{0*} \cdot (a_{0k})^{-1}, C_3)}$, returns $C_{(ID_k, w)} = ((ID^*, w^*), C_0, C_1, C_2^{(2)}, C_3, C_4, C_5, C_6)$ and adds $(ID^*, ID_k, w^*, C_{(ID_k, w^*)}, 0, 0, 1)$ to *RET* and updates *DCT* as $(w^*, K_v, ID_{i+j-k}, ID_k, 0, 1)$.

ii. Else, $\mathcal{C}_1$ recovers $(ID^*, w^*, \perp, a'_{0*}, a'_{1*}, \perp, b'_{3*}, 0)$ from *SKT*, constructs $C_2^{(l)} = \frac{C_2^{(l-1)} \cdot e(a'_{1*} \cdot a_{1k}^{-1}, C_4)}{e(a'_{0*} \cdot b'^{K_v}_{3*} \cdot a_{0k}^{-1}, C_3)}$, returns $C_{(ID_k, w^*)}$, and eventually adds $(ID^*, ID_k, w^*, C_{(ID_k, w^*)}, 0, 0, 1)$ to *RET* and updates *DCT* as $(w^*, K_v, *, ..., ID_{i+j-k}, ID_k, 0, 1)$.

– Else,

i. If $w^* = w$, $ID_{root} = ID^*$, ID_{i+j-k} is in an uncorrupted delegation chain including ID^*, and ID_k is either a corrupted identity or in a corrupted delegation chain, $\mathcal{C}_1$ recovers $(ID^*, w^*, \perp, a_{0*}, a_{1*}, \perp, b_{3*}, 0)$, $(ID^*, w^*, K_v, a'_{0*}, a'_{1*}, \perp, \perp, 0)$ and $(ID_{i+j-k}, w^*, \perp, a_{0i+j-k}, a_{1i+j-k}, \perp, b_{3i+j-k}, 0)$ from *SKT*. If $(ID^*, w^*, K_v, a'_{0*}, a'_{1*}, \perp, \perp, 0) \wedge (ID_{i+j-k}, w^*, \perp, a_{0i+j-k}, a_{1i+j-k}, \perp, b_{3i+j-k}, 0) \notin SKT$, $\mathcal{C}_1$ chooses to construct $sk_{(ID^*, w^*, K_v)}$ and $sk_{(ID_{i+j-k}, w^*)}$, and sets $C_2^{(l)} = \frac{C_2^{(l-1)} \cdot e(a_{0*} \cdot a_{0i+j-k}^{-1} \cdot b_{3i+j-k}^{-K_v}, b_{3*}^{K_v}, C_3)}{e(a_{1*} \cdot a_{1i+j-k}^{-1}, C_4)} \cdot \frac{e(a'_{1*} \cdot (a_{1k})^{-1}, C_4)}{e(a'_{0*} \cdot (a_{0k})^{-1}, C_3)}$, returns the re-encrypted ciphertext and adds $(ID_{i+j-k}, ID_k, w^*, C_{(ID_k, w^*)}, 0, 0, 1)$ to *RET* and updates *DCT* as $(w^*, K_v, ID^*, ..., ID_{i+j-k}, ID_k, 0, 1)$.

ii. Else, $\mathcal{C}_1$ checks whether $(ID_{i+j-k}, w^*, K_v, a_{0i+j-k}, a_{1i+j-k}, \perp, \perp, 0) \in SKT$. If not, $\mathcal{C}_1$ constructs the private key $sk_{ID_{i+j-k}}$ as step (a), derives $sk_{(ID_{i+j-k}, w^*, K_v)} = (a_{0i+j-k}, a_{1i+j-k})$ from $sk_{ID_{i+j-k}}$ and adds the corresponding tuple to *SKT*. Else, $\mathcal{C}_1$ computes $C_2^{(l)} = \frac{C_2^{(l-1)} \cdot e(a_{1i+j-k} \cdot a_{1k}^{-1}, C_4)}{e(a_{0i+j-k} \cdot a_{0k}^{-1}, C_3)}$, outputs the re-encrypted ciphertext and adds $(ID_{i+j-k}, ID_k, w^*, C_{(ID_k, w^*)}, 0, 0, 1)$ to *RET* and updates *DCT* as $(w^*, K_v, *, ..., ID^*, ..., ID_{i+j-k}, ID_k, 0, 1)$.

(d) $\mathcal{O}_{dec}(ID_i, w, C_{(ID_i, w)})$: $\mathcal{C}_1$ verifies Eq. (1). If Eq. (1) does not hold, $\mathcal{C}_1$ outputs $\perp$. Otherwise,

- If $w \neq w^*$:
  i. If $ID_i \neq ID^*$, $\mathcal{C}_1$ checks whether $(ID_i, w, \perp, a_{0i}, a_{1i}, \perp, b_{3i}, *) \in SKT$. If yes, $\mathcal{C}_1$ recovers m by using $sk_{(ID_i, w)}$. Else, $\mathcal{C}_1$ first constructs sk_{ID_i} as step (a), derives $sk_{(ID_i, w)}$ from sk_{ID_i}, and then recovers m by using $sk_{(ID_i, w)}$.
  ii. Else, $\mathcal{C}_1$ checks whether $(ID^*, w, \perp, a_{0*}, a_{1*}, \perp, b_{3*}, 0) \in SKT$. If yes, $\mathcal{C}$ recovers m by using $sk_{(ID^*, w)}$. Else, $\mathcal{C}_1$ constructs $sk_{(ID^*, w)}$, and further recovers m as above.
- If $w = w^*$:
  – If $ID^* = ID_{root}$:
    i. If $ID^* \neq ID_i$, $\mathcal{C}_1$ checks whether $(ID^*, w^*, K_v, a'_{0*}, a'_{1*}, \perp, \perp, 0) \in SKT$ or not. If not, $\mathcal{C}_1$ constructs $sk_{(ID^*, w^*, K_v)} = (a'_{0*}, a'_{1*})$ and adds the corresponding tuple to *SKT*. Else, $\mathcal{C}_1$ recovers $sk_{(ID_i, w^*)} = (a_{0i}, a_{1i}, b_{3i})$ from *SKT*, computes $\sigma = \frac{C_2^{(l)} \cdot e(a_{1*}^{-1} \cdot a_{1i}, C_4)}{e(a_{0*}^{-1} \cdot b_{3*}^{-K_v} \cdot a_{0i} \cdot b_{3i}^{K_v}, C_3)} \cdot \frac{e(a'_{1*}, C_4)}{e(a'_{0*}, C_3)}$, and finally recovers m as in the real scheme.
    ii. Else, $\mathcal{C}_1$ directly uses $sk_{(ID^*, w^*, K_v)}$ to recover m if $(ID^*, w^*, K_v, a'_{0*}, a'_{1*}, \perp, \perp, 0) \in SKT$. Else, $\mathcal{C}_1$ constructs the private key, and then recovers m as above.
  – If $ID^* \neq ID_{root}$:
    i. If $ID^* \neq ID_i$, $\mathcal{C}_1$ recovers m by using $sk_{(ID_i, w^*)}$ if $(ID_i, w^*, \perp, a_{0i}, a_{1i}, \perp, b_{3i}, *) \in SKT$. Else, $\mathcal{C}_1$ constructs the private key, and then recovers m as above.
    ii. Else, $\mathcal{C}_1$ recovers $sk_{(ID^*, w^*)}$ (i.e. $(ID^*, w^*, \perp, a_{0*}, a_{1*}, \perp, b_{3*}, 0)$) from *SKT*, computes $\sigma = \frac{C_2^{(l)} \cdot e(a_{1*}, C_4)}{e(a_{0*} \cdot b_{3*}^{K_v}, C_3)}$, and finally recover m by using σ. Note that the private key is randomly chosen by $\mathcal{C}_1$.
  – If $(ID_j, ID_i, w^*, C_{(ID_i, w^*)}, 0, 0, 1) \in RET$: $\mathcal{C}_1$ recovers m by using $sk_{(ID_i, w^*)}$ if $(ID_i, w^*, \perp, a_{0i}, a_{1i}, \perp, b_{3i}, *) \in SKT$. Else, $\mathcal{C}_1$ constructs the private key, and recovers m as above.

4. **Challenge.** $\mathcal{A}$ outputs m_0 and m_1 to $\mathcal{C}_1$. $\mathcal{C}_1$ flips a random coin $b \in \{0, 1\}$, chooses $\sigma^* \in_R \mathbb{G}_T$, and sets $C_0^* = K_v^*$, $C_1^* = [PRF_{\sigma^*}(h)]^{\lambda_1 - \lambda} \| [PRF_{\sigma^*}(h)]_{\lambda} \oplus m_b$, $C_2^* = \sigma^* \cdot T \cdot e(y_1, h^{\gamma})$, $C_3^* = h$, $C_4^* = h^{\xi + \sum_{i=1}^{3} I_i^* \cdot \gamma_i}$, $C_5^* = h^{w \cdot \gamma_4 + \gamma_5}$, $C_6^* = Sign(K_s^*, (C_1^*, C_3^*, C_4^*, C_5^*))$. The challenge ciphertext is $C_{(ID^*, w^*)}^* = ((ID^*, w^*), C_0^*, C_1^*, C_2^*, C_3^*, C_4^*, C_5^*, C_6^*)$.

If $T = e(g, h)^{\alpha^4}$ (i.e. the input tuple is sampled from $P_{3\text{-wBDHI}^*}$), then $C_{(ID^*, w^*)}^*$ is a valid ciphertext under $I^* = (ID^*, w^*, K_v^*)$. Implicitly letting $h = g^c$ (for an unknown $c \in \mathbb{Z}_q^*$), we have

$$C_1^* = \left[PRF_{\sigma^*}(h)\right]^{\lambda_1 - \lambda} \left\| \left[PRF_{\sigma^*}(h)\right]_{\lambda} \oplus m_b = \left[PRF_{\sigma^*}\left(C_3^*\right)\right]^{\lambda_1 - \lambda} \left\| \left[PRF_{\sigma^*}\left(C_3^*\right)\right]_{\lambda} \oplus m_b \right.\right.,$$
$$C_2^* = \sigma^* \cdot T \cdot e\left(y_1, h^{\gamma}\right) = \sigma^* \cdot e(g, h)^{\alpha^4} \cdot e\left(y_1, h^{\gamma}\right) = \sigma^* \cdot e\left(y_1, y_3 \cdot g^{\gamma}\right)^c = \sigma^* \cdot e(g_1, g_2)^c,$$

$$C_3^* = g^c, \qquad C_4^* = h^{\xi+\sum_{i=1}^{3} I_i^* \cdot \gamma_i} = \left(\prod_{i=1}^{3}\left(g^{\gamma_i}/y_{4-i}\right)^{I_i^*} \cdot g^{\xi} \cdot \prod_{i=1}^{3} y_{4-i}^{I_i^*}\right)^c = \left(h_1^{ID^*} \cdot h_2^{w^*} \cdot h_3^{K_v^*} \cdot g_3\right)^c,$$

$$C_5^* = h^{w\cdot\gamma_4+\gamma_5} = \left(g^{w\cdot\gamma_4+\gamma_5}\right)^c = \left(f_1^w \cdot f_2\right)^c.$$

Otherwise, $T \in_R \mathbb{G}_T$ (i.e. the input tuple is sampled from $R_{3\text{-wBDHI}^*}$), the challenge ciphertext is independent of the bit b in the view of $\mathcal{A}$.

5. **Phase 2.** Same as Phase 1 with the restrictions shown in Definition 2.
6. **Guess.** $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$. If $b' = b$, $\mathcal{C}_1$ outputs 1 (i.e. indicating $T = e(g,h)^{\alpha^4}$); else, it outputs 0 (i.e. indicating $T \in_R G_T$).

Remarks. In Phase 2, due to the capability of recovering C_2^* (which is based on knowledge of $C_2^{(l)}$ and the data stored in *DCT* and *RKT*), $\mathcal{C}_1$ can detect whether a ciphertext $C_{(ID',w^*)}$ is re-encrypted from $C^*_{(ID^*,w^*)}$ or not.[12] This also allows $\mathcal{C}_1$ to check if the ciphertext is mutated by $\mathcal{A}$. Besides, $\mathcal{C}_1$ will reject any query that yields a tuple $(w^*, K_v^*, ID^*, \ldots, 0, 1) \in DCT$.

Let *Forge* be the event that $\mathcal{C}_2$ breaks the security of sUF-CMA of the underlying one-time signature scheme. When the input instance is sampled from $P_{3\text{-wBDHI}^*}$ (i.e. $T = e(g,h)^{\alpha^4}$), the view of $\mathcal{A}$ is identical to the real attacks, and $\mathcal{A}$ has $Pr[\mathcal{C}_1 \to 1 \mid T = e(g,h)^{\alpha^4}] \geq \frac{1}{2} + \epsilon_1 - Pr[Forge \mid T = e(g,h)^{\alpha^4}]$. If $T \in_R \mathbb{G}_T$ (i.e. the instance is sampled from $R_{3\text{-wBDHI}^*}$), m_b is perfectly hidden. $\mathcal{A}$ has $Pr[\mathcal{C}_1 \to 0 \mid T \in_R \mathbb{G}_T] \geq \frac{1}{2} - Pr[Forge \mid T \in_R \mathbb{G}_T]$. In the simulation, either $event_1$ or $event_2$ yields the occurrence of *Forge*. That is, $Pr[Forge] = Pr[event_1 \vee event_2] = Adv^{\mathrm{OTS}}_{\mathcal{C}_2}(1^\lambda)$. Thus, we have

$$\begin{aligned}
2Adv^{D\text{-}3\text{-wBDHI}^*}_{\mathcal{C}_1}(1^\lambda) &= 2\left|Pr\left[\mathcal{C}_1 \to 0 \wedge T = e(g,h)^{\alpha^4}\right] - Pr[\mathcal{C}_1 \to 0 \wedge T \in_R \mathbb{G}_T]\right| \\
&= 2\left|Pr\left[\mathcal{C}_1 \to 0 \mid T = e(g,h)^{\alpha^4}\right]Pr\left[T = e(g,h)^{\alpha^4}\right] - Pr[\mathcal{C}_1 \to 0 | T \in_R \mathbb{G}_T]Pr[T \in_R \mathbb{G}_T]\right| \\
&= \left|Pr\left[\mathcal{C}_1 \to 0 \mid T = e(g,h)^{\alpha^4}\right] - Pr[\mathcal{C}_1 \to 0 \mid T \in_R \mathbb{G}_T]\right| \\
&= \left|1 - Pr\left[\mathcal{C}_1 \to 1 \mid T = e(g,h)^{\alpha^4}\right] - Pr[\mathcal{C}_1 \to 0 \mid T \in_R \mathbb{G}_T]\right| \\
&\geq \left|1 - \left(\frac{1}{2} + \epsilon_1 - Pr\left[Forge \mid T = e(g,h)^{\alpha^4}\right]\right) - \left(\frac{1}{2} - Pr[Forge \mid T \in_R \mathbb{G}_T]\right)\right| \\
&\geq \epsilon_1 - \left(Pr\left[Forge \mid T = e(g,h)^{\alpha^4}\right] + Pr[Forge \mid T \in_R \mathbb{G}_T]\right) \\
&\geq \epsilon_1 - Pr[Forge] \geq \epsilon_1 - Adv^{\mathrm{OTS}}_{\mathcal{C}_2}(1^\lambda).
\end{aligned}$$

Combining the above result and counting for the pseudorandomness of *PRF*, we have $\epsilon_1 \leq 2Adv^{D\text{-}3\text{-wBDHI}^*}_{\mathcal{C}_1}(1^\lambda) + Adv^{\mathrm{OTS}}_{\mathcal{C}_2}(1^\lambda) + Adv^{PRF}_{\mathcal{A}}(1^\lambda)$.

This completes the proof of Theorem 2. □

References

[1] 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.
[2] 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: Lect. Notes Comput. Sci., vol. 4450, Springer, 2007, pp. 201–216.
[3] M. Blaze, G. Bleumer, M. Strauss, Divertible protocols and atomic proxy cryptography, in: K. Nyberg (Ed.), EUROCRYPT, in: Lect. Notes Comput. Sci., vol. 1403, Springer, 1998, pp. 127–144.
[4] D. Boneh, X. Boyen, E.-J. Goh, Hierarchical identity based encryption with constant size ciphertext, in: R. Cramer (Ed.), EUROCRYPT, in: Lect. Notes Comput. Sci., vol. 3494, Springer, 2005, pp. 440–456.
[5] D. Boneh, R. Canetti, S. Halevi, J. Katz, Chosen-ciphertext security from identity-based encryption, SIAM J. Comput. 36 (5) (2007) 1301–1328.
[6] D. Boneh, M.K. Franklin, Identity-based encryption from the weil pairing, SIAM J. Comput. 32 (3) (2003) 586–615.
[7] R. Canetti, S. Hohenberger, Chosen-ciphertext secure proxy re-encryption, in: Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS '07, ACM, New York, NY, USA, 2007, pp. 185–194.
[8] R. Canetti, H. Krawczyk, J.B. Nielsen, Relaxing chosen-ciphertext security, in: D. Boneh (Ed.), CRYPTO, in: Lect. Notes Comput. Sci., vol. 2729, Springer, 2003, pp. 565–582.
[9] 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: Lect. Notes Comput. Sci., vol. 4779, Springer, 2007, pp. 189–202.
[10] R. Cramer, V. Shoup, Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack, SIAM J. Comput. 33 (1) (2003) 167–226.
[11] R.H. Deng, J. Weng, S. Liu, K. Chen, Chosen-ciphertext secure proxy re-encryption without pairings, in: M.K. Franklin, L.C.K. Hui, D.S. Wong (Eds.), CANS, in: Lect. Notes Comput. Sci., vol. 5339, Springer, 2008, pp. 1–17.

[12] If $(ID', w^*, C_{(ID',w^*)})$ is a derivative of $(ID^*, w^*, C^*_{(ID^*,w^*)})$, a part of $C^*_{(ID^*,w^*)}$, i.e. $(ID^*, w^*, C_0^*, C_1^*, C_3^*, C_4^*, C_5^*, C_6^*)$ will be included in $C_{(ID',w^*)}$. Besides, there is a tuple $(w^*, K_v^*, ID^*, \ldots, ID', 1, 0)$ in *DCT*.

[12] L. Fang, W. Susilo, J. Wang, Anonymous conditional proxy re-encryption without random oracle, in: J. Pieprzyk, F. Zhang (Eds.), ProvSec, in: Lect. Notes Comput. Sci., vol. 5848, Springer, 2009, pp. 47–60.
[13] O. Goldreich, S. Goldwasser, S. Micali, How to construct random functions, J. ACM 33 (4) (1986) 792–807.
[14] M. Green, G. Ateniese, Identity-based proxy re-encryption, in: J. Katz, M. Yung (Eds.), ACNS, in: Lect. Notes Comput. Sci., vol. 4521, Springer, 2007, pp. 288–306.
[15] 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.), CT-RSA, in: Lect. Notes Comput. Sci., vol. 7178, Springer, 2012, pp. 349–364.
[16] T. Isshiki, M.H. Nguyen, K. Tanaka, Proxy re-encryption in a stronger security model extended from CT-RSA 2012, in: CT-RSA 2012, in: Lect. Notes Comput. Sci., vol. 7779, Springer, 2013, pp. 277–292.
[17] T. Isshiki, M.H. Nguyen, K. Tanaka, Factoring-based proxy re-encryption schemes, in: W. Susilo, R. Reyhanitabar (Eds.), ProvSec, in: Lect. Notes Comput. Sci., vol. 8209, Springer, 2013, pp. 309–329.
[18] A.-A. Ivan, Y. Dodis, Proxy cryptography revisited, in: NDSS, The Internet Society, 2003.
[19] K. Liang, Z. Liu, X. Tan, D.S. Wong, C. Tang, A CCA-secure identity-based conditional proxy re-encryption without random oracles, in: Proceedings of the 15th ICISC, in: Lect. Notes Comput. Sci., vol. 7839, Springer, 2013, pp. 231–246.
[20] B. Libert, D. Vergnaud, Unidirectional chosen-ciphertext secure proxy re-encryption, in: R. Cramer (Ed.), Public Key Cryptography, in: Lect. Notes Comput. Sci., vol. 4939, Springer, 2008, pp. 360–379.
[21] S. Luo, Q. Shen, Z. Chen, Fully secure unidirectional identity-based proxy re-encryption, in: H. Kim (Ed.), ICISC, in: Lect. Notes Comput. Sci., vol. 7259, Springer, Heidelberg, 2011, pp. 109–126.
[22] M. Mambo, E. Okamoto, Proxy cryptosystems: delegation of the power to decrypt ciphertexts, IEICE Trans. E80-A (1) (1997) 54–63.
[23] T. Matsuda, R. Nishimaki, K. Tanaka, CCA proxy re-encryption without bilinear maps in the standard model, in: P.Q. Nguyen, D. Pointcheval (Eds.), Public Key Cryptography, in: Lect. Notes Comput. Sci., vol. 6056, Springer, 2010, pp. 261–278.
[24] T. Matsuo, Proxy re-encryption systems for identity-based encryption, in: T. Takagi, T. Okamoto, E. Okamoto, T. Okamoto (Eds.), Pairing, in: Lect. Notes Comput. Sci., vol. 4575, Springer, 2007, pp. 247–267.
[25] T. Mizuno, H. Doi, Secure and efficient IBE-PKE proxy re-encryption, IEICE Trans. 94-A (1) (2011) 36–44.
[26] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: R. Cramer (Ed.), EUROCRYPT, in: Lect. Notes Comput. Sci., vol. 3494, Springer, 2005, pp. 457–473.
[27] J. Shao, Z. Cao, Multi-use unidirectional identity-based proxy re-encryption from hierarchical identity-based encryption, Inf. Sci. 206 (0) (2012) 83–95.
[28] Q. Tang, Type-based proxy re-encryption and its construction, in: D.R. Chowdhury, V. Rijmen, A. Das (Eds.), INDOCRYPT, in: Lect. Notes Comput. Sci., vol. 5365, Springer, 2008, pp. 130–144.
[29] Q. Tang, P.H. Hartel, W. Jonker, Inter-domain identity-based proxy re-encryption, in: M. Yung, P. Liu, D. Lin (Eds.), Inscrypt, in: Lect. Notes Comput. Sci., vol. 5487, Springer, 2008, pp. 332–347.
[30] H. Wang, Z. Cao, L. Wang, Multi-use and unidirectional identity-based proxy re-encryption schemes, Inf. Sci. 180 (20) (2010) 4042–4059.
[31] L. Wang, L. Wang, M. Mambo, E. Okamoto, Identity-based proxy cryptosystems with revocability and hierarchical confidentialities, IEICE Trans. 95-A (1) (2012) 70–88.
[32] L. Wang, L. Wang, M. Mambo, E. Okamoto, New identity-based proxy re-encryption schemes to prevent collusion attacks, in: M. Joye, A. Miyaji, A. Otsuka (Eds.), Pairing, in: Lect. Notes Comput. Sci., vol. 6487, 2010, pp. 327–346.
[33] B. Waters, Efficient identity-based encryption without random oracles, in: R. Cramer (Ed.), EUROCRYPT, in: Lect. Notes Comput. Sci., vol. 3494, Springer, 2005, pp. 114–127.
[34] B. Waters, Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions, in: S. Halevi (Ed.), CRYPTO, in: Lect. Notes Comput. Sci., vol. 5677, Springer, 2009, pp. 619–636.
[35] 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.
[36] J. Weng, Y. Zhao, G. Hanaoka, On the security of a bidirectional proxy re-encryption scheme from PKC 2010, in: D. Catalano, N. Fazio, R. Gennaro, A. Nicolosi (Eds.), Public Key Cryptography, in: Lect. Notes Comput. Sci., vol. 6571, Springer, 2011, pp. 284–295.
[37] J. Weng, Y. Yang, Q. Tang, R.H. Deng, F. Bao, Efficient conditional proxy re-encryption with chosen-ciphertext security, in: P. Samarati, M. Yung, F. Martinelli, C.A. Ardagna (Eds.), ISC, in: Lect. Notes Comput. Sci., vol. 5735, Springer, 2009, pp. 151–166.
[38] J. Weng, R.H. Deng, X. Ding, C.-K. Chu, J. Lai, Conditional proxy re-encryption secure against chosen-ciphertext attack, in: W. Li, W. Susilo, U.K. Tupakula, R. Safavi-Naini, V. Varadharajan (Eds.), ASIACCS, ACM, 2009, pp. 322–332.