PRE$^+$: Dual of Proxy Re-encryption for Secure Cloud Data Sharing Service

Xu An Wang*

School of Telecommunications Engineering, Xidian University, P. R. China
Key laboratory of Information and Network Security, Engineering University of Chinese Armed Police Force, P. R. China
Email: wangxazjd@163.com
*Corresponding author

Fatos Xhafa

Department of Computer Science, Technical University of Catalonia, Spain
Email: fatos.xhafa@gmail.com

Jianfeng Ma

School of Cyber Engineering, Xidian University, P. R. China
Email:jfma@mail.xidian.edu.cn

Yunlong Ge

Officer College of Chinese Armed Police Force, P. R. China
Email:339020145@qq.com

Abstract: With with the rapid development of very large, diverse, complex, and distributed datasets generated from internet transactions, emails, videos, business information systems, manufacturing industry, sensors and internet of things etc., cloud and big data computation have emerged as a cornerstone of modern applications. Indeed, on the one hand, cloud and big data applications are becoming a main driver for economic growth. On the other hand, cloud and big data techniques may threaten people and enterprises' privacy and security due to ever increasing exposure of their data to massive access. In this paper, aiming at providing secure cloud data sharing services in cloud storage, we propose a scalable and controllable cloud data sharing framework for cloud users (called: **Scanf**). To this end, we introduce a new cryptographic primitive, namely, PRE$^+$, which can be seen as the dual of traditional proxy re-encryption (PRE) primitive. All the traditional PRE schemes until now require the *delegator* (or the delegator and the *delegatee* cooperatively) to generate the re-encryption keys. We observe that this is not the only way to generate the re-encryption keys, *the encrypter* also has the ability to generate re-encryption keys. Based on this observation, we construct a new PRE$^+$ scheme, which is almost the same as the traditional PRE scheme except the re-encryption keys generated by *the encrypter*. Compared with PRE, our PRE$^+$ scheme can easily achieve *the non-transferable property and message-level based fine-grained delegation*. Thus our **Scanf** framework based on PRE$^+$ can also achieve these two properties, which is very important for users of cloud storage sharing service. We also roughly evaluate our PRE$^+$ scheme's performance and the results show that our scheme is efficient and practica for cloud data storage applications.

Keywords: Secure cloud data sharing service; Dual of proxy re-encryption; Non-transferable property; Message-level based fine-grained delegation.

Reference to this paper should be made as follows: Wang, X., Ma, J., Fatos, L. Barolli, X., Ge Y., 'PRE$^+$: Dual of Proxy Re-encryption for Secure Cloud Data Sharing Service', *Int. J. Int. J. of Web and Grid Services*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Xu An Wang now is an associate professor in the Engineering University of Chinese Armed Police Force. His main research interests include public key cryptography and cloud security.

Fatos Xhafa now is a professor in Technical University of Catalonia. His main research interests include cloud computation and big data analysis. He has published about 200 papers in the field of grid, internet, cloud and big data computation.

Jianfeng Ma now is a professor in Xidian University. He is also a Yangtze River scholar in China. His main research interests include cyber engineering security. He has published about 150 papers in the field of public key cryptography and cloud security.

Yunlong Ge now is a lecturer in Officer College of Chinese Armed Police Force. His main research interest are public key cryptography and provable security.

1 Introduction

Nowadays cloud computation has become commonplace for data storage, processing, service sharing for individuals, institutions, and enterprises world wide. As regards the computation and processing, the core novelty of this technology is its ability of enabling heavy computation tasks, which can be distributed to large slave computation nodes under the management of master computation nodes, like the system structure of Map-Reduce. Scalability in its vertical and horizontal dimensions, is the essential property of cloud computation, while enterprises can maintain their data warehouse and run information systems on it in the cloud just like running the local information systems. Cloud computing is also referred to as 5th utility computing as it enables end-users to use and pay the cloud storage/computation service just like using gas, water and electrical power utilities. Generally speaking, in the Cloud services are offered through the paradigm of “Everything as a Aservice", an examples of this are Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS) and Data-as-a-Service (DaaS), to name a few.

1.1 Proxy Re-encryption for Secure Cloud Data Sharing Service

Here we focus on a very common kind of cloud service: Data-as-a-Service (DaaS), aimed at providing convenient data storage and sharing service for data owners. Data owners like individuals, institutions and enterprises can outsource their data to the cloud server without maintaining the local data copy and the complicated software/hardware management. However, when data owners outsource their data to the cloud, they run some risks on the privacy and security of their data hosted at cloud data centres. Thus, it is often a reasonable practice to first encrypt the data locally and then outsource the ciphertexts to the cloud. However, although in this way the outsourced data can be secure, but it becomes more difficult to operate with the data such as sharing the encrypted data, processing and mining, etc..

In order to enable sharing data among cloud data users, recently researchers have proposed a new data sharing framework based on a cryptographic primitive named proxy re-encryption. Proxy re-encryption (PRE), allows a semi-trusted proxy to transform a ciphertext originally intended for Alice into the one which can be decrypted by Bob. The proxy in PRE is given a re-encryption key with which he/she can perform the conversion without knowing the corresponding plaintexts. A PRE scheme is *bidirectional* if the proxy can use the re-encryption key to divert ciphertexts from Alice to Bob and vice versa. Otherwise, it is called *unidirectional*. According to how the conversion is performed, PRE can be classified as *multi-hop* and *single-hop*. A multi-hop PRE scheme supports “re-encryption chain", namely any ciphertexts converted by the proxy during the re-encryption phase can be re-encrypted to the ciphertexts of someone else. In other words, the re-encryption phase in multi-hop PRE can be conducted as many times as needed. This differs from a single-hop PRE scheme, where ciphertexts converted by the proxy cannot be re-encrypted again.

1.2 The Dilemma

We can roughly see how to use PRE for secure cloud data sharing service and its dilemma from Fig. 1. First we describe a typical scenario for secure sharing cloud storage by using PRE as follows:

1. Data owner Company TaoBao want to use cloud services for managing its very large business related data sets, such as the consumers' preference, products selling prices, analytics of logistics for goods etc. These data sets are very sensitive and the Company TaoBao is not willing to outsource them in plaintext form. Thus, it prefers to first encrypt the data sets $file$ by using block cipher such as AES and then encrypt the block cipher key K by using public key $TaoBao$, after that it outsources the ciphertexts to the cloud.

2. Later, the Company TaoBao signs a contract on data sharing with another business, the Company 360, for evaluating its information system and data warehouse's security. Certainly the Company TaoBao is not willing to directly share its secret key sk_{TaoBao} with the Company 360. Proxy re-encryption is a good mechanism for this purpose. Company TaoBao just computes the re-encryption key $rk_{TaoBao \to 360}$ by using its secret key and the Company 360's public key, and outsources it to the cloud.

3. The cloud re-encrypts the ciphertext from $(E_K(file)||E_{TaoBao}(K))$ to be $(E_K(file)||E_{360}(K))$ and sends them to data user Company 360. After obtaining these ciphertexts, it can decrypt them by using his own secret key and thus can run its evaluation. Furthermore, data owner Company TaoBao can use conditional proxy re-encryption, which is a variant of PRE for fine-grained control on the sharing content for the Company 360.

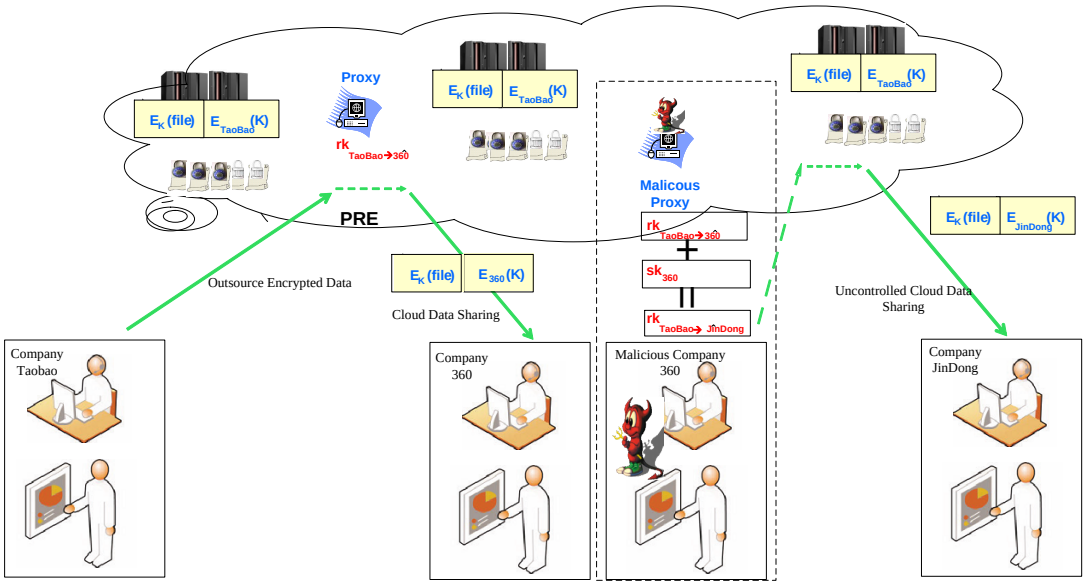

Figure 1 PRE for secure cloud data sharing service and its dilemma

Almost all existing frameworks of PRE for secure cloud data sharing service follow the above paradigm. However, there is a dilemma these proposals have to face to, which can be described as follows:

1. Suppose the Company JinDong, which is a competitor of Company TaoBao, wants to know some critical contents of Company TaoBao's data sets. It corrupts both the cloud and Company 360 by paying large amount of money.

2. The cloud and Company 360 can collude to derive a new re-encryption key $rk_{TaoBao \to JinDong}$ from the old re-encryption key $rk_{TaoBao \to 360}$ and Company 360's secret key sk_{360}.

3. By using the new re-encryption key $rk_{TaoBao \to JinDong}$, the cloud can transform ciphertext from $(E_K(file)||E_{TaoBao}(K))$ to be $(E_K(file)||E_{JinDong}(K))$.

4. After obtaining $(E_K(file)||E_{JinDong}(K))$, Company JinDong can decrypt the ciphertexts to obtain the data content $file$, which is unpredicted by Company TaoBao.

One might wonder if Company 360 can directly send the data content to Company JinDong, why it first colludes with the cloud to derive the new re-encryption key, and then the cloud implements the ciphertext transformation? The reason lies in the auditing capability: in the first case, Company 360 needs to be online always and thus can easily be traced; while in the second case, Company 360 can be always offline and untraceable, and the cloud can claim it is innocent on the new re-encryption key.

1.3 Our Contribution

Aiming at solving this dilemma, we propose a new primitive PRE^+ and a new framework **Scanf** for secure data sharing service at cloud. Our framework can be employed as a middleware component for effective handling data storage on demand with organizational separation of the provider and consumer for the newly rapidly emerging DaaS cloud service. Separating data usage from the software to access the data is a very common practice but it is at the same time a challenging problem for modern DaaS cloud service. By employing the PRE^+ primitive and the **Scanf** framework, the cloud can achieve scalable control on which data consumer can access the encrypted the data. Furthermore, this access can be controlled without worrying about further potential leakage of the data by the data consumers, for this leakage can be audited. Concretely our contributions are the following:

1. We first point out the dilemma faced by almost all existing secure cloud storage sharing service based on proxy re-encryption, that is, the malicious delegatee and proxy can collude to re-delegate the re-encryption right to other users, which can be denoted as the transferable delegation problem.
2. We introduce a new primitive PRE^+ which can be seen as dual of traditional proxy re-encryption, where we also discuss some interesting properties such as non-transferability and message-based fine-grained control on the delegation. We further analyse why this primitive can be used to solve the transferable delegation problem.
3. We give the definition and security model PRE^+. We also construct a concrete chosen ciphertext secure PRE^+ scheme in the random oracle model and prove its security as well as analyse its features and performance.
4. Finally, we propose a **s**calable and **c**ontrollable cloud d**a**ta shari**n**g **f**ramework for cloud users, denoted **Scanf**, based on our PRE^+ scheme and provide a rough analysis of its security.

1.4 Related Work

The first PRE scheme was proposed by Blaze *et al*. in Eurocrypt'98 [7]. But due to some undesirable properties of this scheme such as *bidirectional* and *collusion-unresistant* (the delegatee and the proxy can collude to get the delegator's secret key), it did not receive much attention from researchers. In NDSS'05, Ateniese *et al*. proposed new unidirectional PRE schemes, discussed nine desirable properties for PRE, namely, *unidirectional, non-interactive, proxy invisibility, original-access, key optimal, collusion-safe, temporary, non-transitive, non-transferable* and finally explored its potential applications especially in distributed secure file systems [1, 2]. Since then, PRE attracts more and more attention from the researchers in the field. As a matter of fact, many interesting results about PRE have been achieved. From the point view of theoretic construction, different versions of PRE have been proposed, such as CCA-secure PRE [8, 13, 30, 33, 34, 49, 48, 35], PRE in the identity based setting [11, 16, 32], PRE in the attribute based setting [29], PRE in the broadcast setting [12], conditional PRE or type-based PRE [45, 12, 46, 39, 21], key private PRE [3], PRE with keyword search [36] etc. From the point view of practical application, PRE has found many interesting applications, such as in encrypted e-mail forwarding [8], key escrow [14], distributed secure file systems [1, 2], security in publish/subscribe systems [25], multicast [10], secure certified email mailing lists [23, 24], interoperable architecture of DRM [40], access control [41], privacy for public transformation [17], and securely obfuscating re-encryption [19] etc. Recently researchers are showing increased interest on using PRE for secure cloud storage sharing. Liang *et al*. [27] first proposed the notion of deterministic finite automata-based functional proxy re-encryption, which combines the feature of DFA-based functional encryption with proxy re-encryption and discuss its application in secure public cloud data sharing. Later, they also [27] proposed a privacy-preserving ciphertext multi-sharing control for big data storage mechanism, which is based on the cryptographic primitive of anonymous multi-hop identity based conditional proxy re-encryption AMH-IBCPRE scheme. Shao et *al*. [38] discussed how to use proxy re-encryption for secure mobile cloud service, they designed a framework, which can

support fine-grained privacy-preserving location-based service framework for mobile devices. At Infocom15 [37], they designed a fine-grained data sharing mechanism in cloud computing for mobile devices.

1.5 Paper Organization

The rest of the paper is organized as follows. In Section 2, we start by informally introducing our new primitive PRE$^+$ and a rough analysis on its properties. In Section 3, we give some preliminaries which are necessary for understanding our concrete construction for PRE$^+$. In Section 4, we give the definition and security notions for PRE$^+$. In Section 5, we give a concrete single-hop PRE$^+$ scheme and prove its security as well as its extension to multi-hop variants. We also roughly analyse our scheme's performance and features. In Section 6, we propose a framework Scanf for secure sharing data in cloud storage. Finally, we conclude our paper with some interesting open problems in Section 7.

2 A New Primitive PRE$^+$: Dual of Proxy Re-encryption

The rationale for proposing a new scheme is based on the following observation. We observe that all the PRE schemes until now require the delegator (or the delegator and the delegatee cooperatively) to generate the re-encryption keys, which can be seen in Fig. 2. However this is not the only way to generate the re-encryption keys. *The encrypter* also has this ability, after all, it has the full control on his plaintext and its corresponding ciphertext. We exploit the possibility of constructing such a PRE scheme, which we denote PRE as PRE$^+$. A graphical description on PRE$^+$ can be seen in Fig. 3.

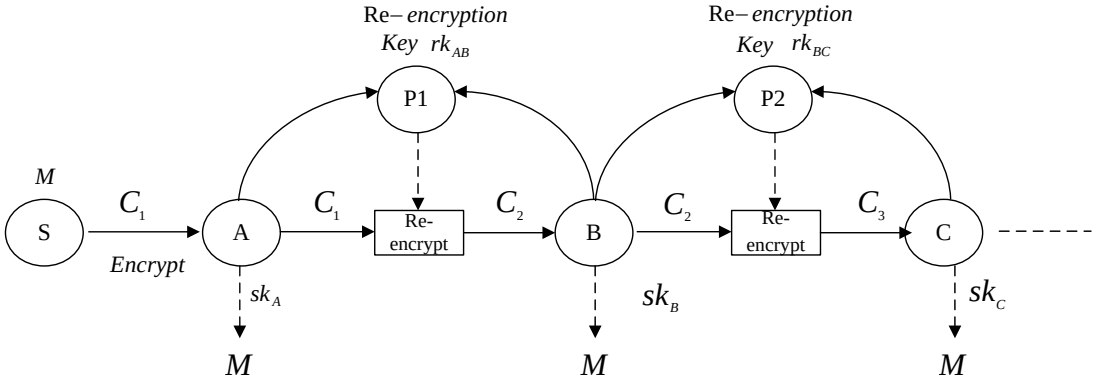

Figure 2 Traditional proxy re-encryption

Intuitively, the PRE$^+$ is the dual of the traditional PRE. In the traditional PRE, the receiver of the ciphertexts (delegator) delegates the decryption right to the delegatee. While in the new PRE$^+$, encrypter-the sender of the ciphertexts delegates the decryption right to the delegatee. Naturally PRE$^+$ can be categorized as the *single-hop* and *multi-hop* variants. At first sight, the PRE$^+$ seems to be not useful for practical applications. The encrypter can directly encrypt the message to the delegatee, why there is needed for PRE$^+$ to delegate the decryption rights? The answer lies in the following facts: First, this delegation can considerably improve the computation and communication efficiency compared with the twice encryption method. Secondly, the PRE$^+$ can solve some long standing open problems in PRE like delegation transferability and fine-grained delegation. Thirdly, considering the multi-hop variants of PRE$^+$, it runs like broadcast encryption except it relies on some semi-trusted proxies. And these semi-trusted proxies can be easily found in modern networks. Thus it can be well suited for some particular applications. Finally, PRE$^+$ schemes also have many interesting applications like secure cloud computation and multicast, etc. Compared with the traditional PRE, PRE$^+$ has the following advantages:

- *Fine-grained delegation.* For PRE, all the proposed schemes so far can only achieve time-based (such as [1, 2, 30, 16]) or condition-based (such as [45, 12, 46, 39, 21]) fine-grained delegation. But this situation is not

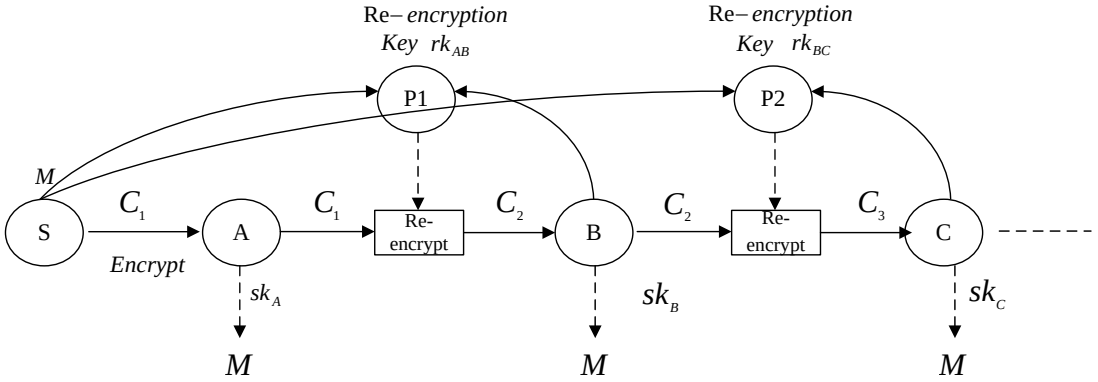

Figure 3 PRE^+: Dual of proxy re-encryption

fully satisfactory, for the proxy in these schemes can still re-encrypt the delegator's ciphertext *infinitely* times if and only if it is in the valid time or with the valid condition. Many practical applications *do not* need this property as it is preferable that the delegator can control on the messages the proxy can re-encrypt. Using our PRE^+ scheme this problem can be easily solved. Indeed, in PRE^+, the value that can control the delegation is the encrypter's ephemeral randomness. If we follow this principle, then the encrypter needs to use the same ephemeral randomness for the delegated messages. If the encrypter wants to allow the proxy to re-encrypt the message, it encrypts the message with the "same" ephemeral randomness. Otherwise it encrypts the message with a "different" ephemeral randomness. Thus, we can see that the encrypter can easily achieve message-based fine-grained control on the delegation.

- *Non-transferable delegation.* Non-transferable delegation means that, if the delegator $\mathcal{A}$ delegates the re-encryption right for the delegatee $\mathcal{B}$ to the proxy $\mathcal{P}$, $\mathcal{P}$ and $\mathcal{B}$ can not collude to delegate, representing $\mathcal{A}$, new re-encryption right for any other user. For PRE, all the schemes until now cannot solve this problem naturally. For example, in [30], the authors remarked that constructing systems withstanding the *transfer of delegation* attacks remained as open problems. Although some researchers try to solve this open problem [44, 43, 18] in the identity based setting, but their proposals heavily rely on the PKG, which is not a very natural solution. Again in our PRE^+ scheme this problem can be solved very naturally. In the PRE^+, the encrypter computes the re-encryption key by using his ephemeral randomness for the messages and the delegatee's public key. The normal ciphertext receiver $\mathcal{A}$ will now be independent with the delegatee $\mathcal{B}$ and the proxy $\mathcal{P}$, now if $\mathcal{B}$ and $\mathcal{P}$ can not collude to derive the ephemeral randomness, it will be non-transferable.

The multi-hop PRE^+ runs likely the broadcast encryption (BE), for both primitives can be viewed as the encryption primitive dealing with one encrypter but many receivers. We here also discuss the difference between PRE^+ and BE. Obviously, multi-hop PRE^+ relies heavily on the physical objects, the semi-trusted proxies. But we remark that PRE^+ has its own advantages compared with BE for some particular applications.

- *Scalability.* Generically speaking, BE is a primitive tailorable to the scenario of one encrypter (encrypting messages) to many receivers. When designing the BE schemes, researchers focused on its efficiency including the communication cost such as the ciphertext length, public/private key size and the computation cost, while payed little attention on its scalability. When a new user joins or an old user leaves the receiver set, in some cases, the encrypter has to compute and send the ciphertext again for the whole receiver set or the whole system setup has to be rebooted. For multi-hop PRE^+, the scalability issue can be solved smoothly. If a new user joins or an old user leaves the receiver set, what is needed to do is the delegator (encrypter) adjust the new user's nearest proxy's re-encryption key, which is a very easy task.

- *Efficiency.* In current BE schemes, researchers always have to make some trade-off between size of the ciphertext and public/private keys, for example, The best known fully collusion resistant BE scheme is the BGW scheme, proposed by Boneh, Gentry and Waters in Eurocrypt'05 [5], which can achieve $\mathcal{O}(\sqrt{n})$-size ciphertexts and

public key; others chemes can achieve constant size ciphertext, $\mathcal{O}(n)$-size public key and constant size private keys. While in the PRE^+ scheme, the constant size of ciphertext and public/private keys can be easily achieved.

3 Preliminaries

3.1 Bilinear Map

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two groups of order p for some large prime p. A bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ between these two groups satisfies the following properties:

- Bilinear: We say that a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is bilinear if $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p$.
- Non-degenerative: The map does not send all pairs in $\mathbb{G} \times \mathbb{G}$ to the identity in $\mathbb{G}_T$. We observe that $\mathbb{G}$, $\mathbb{G}$ are groups of prime order, this implies that if P is a generator of $\mathbb{G}$ then $e(P, P)$ is a generator of $\mathbb{G}_T$.
- Computable: There is an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in \mathbb{G}$.

We denote BSetup as an algorithm that, on input the security parameter 1^k, it outputs the parameters for a bilinear map as $(p, g, \mathbb{G}, \mathbb{G}_T, e)$, where $p \in \Theta(2^k)$.

3.2 Complexity Assumptions

Let $(p, g, \mathbb{G}, \mathbb{G}_T, e)$ be a bilinear group of prime order p and P be its generator. Here we define the extended decisional bilinear Diffie-Hellman (eDBDH) assumption, extended decisional linear (eDL) assumption and variant of discrete logarithm (vDL) assumption.

eDBDH Problem. Let $(p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathsf{BSetup}(1^k)$ The eDBDH problem is as follows: given given $(g, g^a, g^{ab}, g^{cd/b}, g^{cde/b}, g^{cf}, g^{b/e}, g^{ued/b}g^{sf}, T)$ or random. for some $a, b, c, d, e \in Z_p^*$ and $T \in \mathbb{G}_T$, decide whether $T = e(g, g)^{ad}$. An algorithm $\mathcal{A}$ has advantage ϵ in solving eDBDH problem if

$$|Pr[\mathcal{A}(g, g^a, g^{ab}, g^{cd/b}, g^{cde/b}, g^{cf}, g^{b/e}, g^{ued/b}g^{sf}, e(g, g)^{ad}) = 0]$$
$$-Pr[\mathcal{A}(g, g^a, g^{ab}, g^{cd/b}, g^{cde/b}, g^{cf}, g^{b/e}, g^{ued/b}g^{sf}, T) = 0]| \geq \epsilon$$

where the probability is over the random choices of a, b, c, d, e, u, f, s in Z_p^*, the random choices of T in $\mathbb{G}_T$, the random choice of $g \in \mathbb{G}^*$, and the random bits of $\mathcal{A}$.

Definition 3.1: (eDBDH Assumption) *We say that the ϵ-eDBDH assumption holds if no PPT algorithm has advantage at least ϵ in solving the eDBDH problem.*

eDL Problem. Let $(p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow BSetup(1^k)$. The eDL problem is as follows: Given $(g, g^{\alpha_2}, g^{\alpha_1\alpha_2\alpha_4}, g^{\alpha_3}, g^{\alpha_1\alpha_3\alpha_5}, g^{\alpha_1\alpha_4 u})$, decide whether $T = g^{\alpha_5 - \alpha_4}$. An algorithm $\mathcal{A}$ has advantage ϵ in solving eDL problem if

$$|Pr[\mathcal{A}(g, g^{\alpha_2}, g^{\alpha_1\alpha_2\alpha_4}, g^{\alpha_3}, g^{\alpha_1\alpha_3\alpha_5}, g^{\alpha_1\alpha_4 u}, g^{\alpha_5 - \alpha_4}) = 0]$$
$$-Pr[\mathcal{A}(g, g^{\alpha_2}, g^{\alpha_1\alpha_2\alpha_4}, g^{\alpha_3}, g^{\alpha_1\alpha_3\alpha_5}, g^{\alpha_1\alpha_4 u}, T) = 0]| \geq \epsilon$$

where the probability is over the random choices of $\alpha_1, \alpha_2, \cdots, \alpha_5, u$ in Z_p^*, the random choices of T in $\mathbb{G}_T$, the random choice of $g \in \mathbb{G}$, and the random bits of $\mathcal{A}$.

Definition 3.2: (eDL Assumption) *We say that the ϵ-eDL assumption holds if no PPT algorithm has advantage at least ϵ in solving the eDL problem.*

vDL Problem. Let $(p, g, \mathbb{G}, \mathbb{G}_T, e) \leftarrow BSetup(1^k)$. The vDL problem is as follows: Given $(g^{\beta_1}, g^{u\beta_2/\beta_1}, g^{\beta_2-\beta_3}, g^{\beta_2/\beta_1})$, compute (β_2, β_1), An algorithm $\mathcal{A}$ has advantage ϵ in solving vDL problem if

$$|Pr[\mathcal{A}(g^{\beta_1}, g^{u\beta_2/\beta_1}, g^{\beta_2-\beta_3}, g^{\beta_2/\beta_1}) = (\beta_2, \beta_1)]| \geq \epsilon$$

where the probability is over the random choices of $\beta_1, \beta_2, \beta_3, u$ in Z_p^*, the random choice of $g \in \mathbb{G}$, and the random bits of $\mathcal{A}$.

Definition 3.3: (vDL Assumption) *We say that the ϵ-vDL assumption holds if no PPT algorithm has advantage at least ϵ in solving the vDL problem.*

4 Definition and Security Model for PRE$^+$

4.1 Definition

Definition 4.1: *PRE$^+$ (Dual of proxy re-encryption) scheme is a tuple of algorithms* $\mathsf{PRE}^+.\mathsf{KeyGen}$, $\mathsf{PRE}^+.\mathsf{ReKeyGen}$, $\mathsf{PRE}^+.\mathsf{Enc}$, $\mathsf{PRE}^+.\mathsf{ReEnc}$, $\mathsf{PRE}^+.\mathsf{Dec}$)

$\mathsf{PRE}^+.\mathsf{KeyGen}(1^k) \rightarrow (pk, sk)$: On input the security parameter 1^k, this algorithm outputs the decrypter's public key/secret key pair (pk_1, sk_1) and the delegatee's public/secret key pair (pk_2, sk_2). Here we emphasis that in PRE$^+$, the delegator is the encrypter, while in traditional PRE the delegator is the decrypter (ciphertext receiver).

$\mathsf{PRE}^+.\mathsf{Enc}(pk_1, m) \rightarrow C_2$: On input a public key pk_1, ephemeral randomness consisting of (r, r') [1], a message $m \in \{0,1\}^n$, this algorithm (the encrypter) outputs a second-level ciphertext C_2.

$\mathsf{PRE}^+.\mathsf{ReKeyGen}(r', pk_1, pk_2) \rightarrow rk_{1\rightarrow 2}$: On input ephemeral randomness r' for pk_1, public key pk_2, this algorithm (the encrypter) outputs a re-encryption key $rk_{1\rightarrow 2}$.

$\mathsf{PRE}^+.\mathsf{ReEnc}(rk_{1\rightarrow 2}, C_2) \rightarrow C_1$: On input a re-encryption key $rk_{1\rightarrow 2}$ and a ciphertext C_2, this algorithm (the proxy) outputs a first-level ciphertext C_1 or the error symbol $\perp$.

$\mathsf{PRE}^+.\mathsf{Dec}_2(sk_2, C_2)$: On input a secret key sk_2 and a second-level ciphertext C_2, this algorithm (the decrypter) outputs a message $m \in \{0,1\}^n$ or $\perp$.

$\mathsf{PRE}^+.\mathsf{Dec}_1(sk_1, C_1)$: On input a secret key sk_1 and a first-level ciphertext C_1, this algorithm (the delegatee) outputs a message $m \in \{0,1\}^n$ or $\perp$.

Roughly speaking, the correctness requires that, for all $(pk_i, sk_i) \leftarrow \mathsf{PRE}^+.\mathsf{KeyGen}(1^k)$ and $(pk_j, sk_j) \leftarrow \mathsf{PRE}^+.\mathsf{KeyGen}(1^k)$, it holds that

$$\mathsf{PRE}^+.\mathsf{Dec}_2(sk_i, \mathsf{PRE}^+.\mathsf{Enc}(pk_i, m)) = m.$$
$$\mathsf{PRE}^+.\mathsf{Dec}_1(sk_j, \mathsf{PRE}^+.\mathsf{ReEnc}(\mathsf{PRE}^+.\mathsf{ReKeyGen}(r', pk_1, pk_2), \mathsf{PRE}^+.\mathsf{Enc}(pk_i, m)) = m.$$

4.2 Security Models

We start by giving some definitions.

Definition 4.2: (CCA-security for the Second-level Ciphertext). *A single-use unidirectional* PRE^+ *scheme is CCA-secure for the second-level ciphertext if the advantage of any PPT adversary $\mathcal{A}$ in the following game played between a challenger $\mathcal{C}$ and $\mathcal{A}$ is negligible in the security parameter k. Note that we work in the static corruption model, where the adversary should decide the corrupted users before the game starts.*

Setup: The Challenger sets up the system parameters.

Phase 1: The adversary $\mathcal{A}$ adaptively issues queries $q_1, \cdots, q_{n_1}$ where query q_i is one of:

- $\mathcal{O}_{pk}$: On input an index i, the Challenger takes a security parameter k, and responds by running algorithm $\mathsf{PRE}^+.\mathsf{KeyGen}(1^k)$ to generate a key pair (pk_i, sk_i), gives pk_i to $\mathcal{A}$ and records (pk_i, sk_i) in table T_K.
- $\mathcal{O}_{sk}$: On input pk_i by $\mathcal{A}$, where pk_i is from $\mathcal{O}_{pk}$, if pk_i is corrupted, the Challenger searches pk_i in table T_K and returns sk_i; otherwise, the Challenger returns $\perp$.
- $\mathcal{O}_{rk}$: On input (pk_i, pk_j) by $\mathcal{A}$, where pk_i, pk_j are from $\mathcal{O}_{pk}$, the Challenger returns the re-encryption key $rk_{pk_i \to pk_j} = \mathsf{PRE}^+.\mathsf{ReKeyGen}(r', pk_i, pk_j)$, where sk_i is the secret key corresponding to pk_i.
- $\mathcal{O}_{re}$: On input (pk_i, pk_j, C) by $\mathcal{A}$, where pk_i, pk_j are from $\mathcal{O}_{pk}$, the Challenger returns the re-encrypted ciphertext $C' = \mathsf{PRE}^+.\mathsf{ReEnc}(\mathsf{PRE}^+.\mathsf{ReKeyGen}(r', pk_i, pk_j), C)$, where sk is the secret key corresponding to pk.
- $\mathcal{O}_{dec}$: On input (pk_i, C_i), where pk_i is from $\mathcal{O}_{pk}$, the Challenger returns $\mathsf{PRE}^+.\mathsf{Dec}_1(sk_i, C_i)$ or $\mathsf{PRE}^+.\mathsf{Dec}_2(sk_i, C_i)$, where sk_i is the secret key corresponding to pk_i.

Challenge: Once the adversary $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length plaintexts m_0, m_1 from the message space, and a public key pk^* on which it wishes to challenge. There are three constraints on the public key pk^*, (i) it is from $\mathcal{O}_{pk}$; (ii) it is uncorrupted; (iii) if $(pk^*, \bigstar)$ did appear in any query to $\mathcal{O}_{rk}$, then $\bigstar$ is uncorrupted. The Challenger picks a random bit $\mathbf{b} \in \{0, 1\}$ and sets

$$C^* = \mathsf{PRE}^+.\mathsf{Enc}(pk^*, m_{\mathbf{b}})$$

It sends C^* as the challenge to $\mathcal{A}$.

Phase 2: The adversary $\mathcal{A}$ adaptively issues more queries $q_{n_1+1}, \cdots, q_n$, but with the following restrictions.

- $\mathcal{O}_{rk}$: On input (pk_i, pk_j) by $\mathcal{A}$, the following requirements should be all satisfied.
  - pk_i and pk_j are from $\mathcal{O}_{pk}$;
  - if $pk_i = pk^*$, then pk_j is uncorrupted.
- $\mathcal{O}_{re}$: On input (pk_i, pk_j, C_i) by $\mathcal{A}$, the following requirements should be all satisfied.
  - pk_i and pk_j are from $\mathcal{O}_{pk}$;
  - if (pk_i, C_i) is a derivative, the Challenger outputs $\perp$. Otherwise the challenger returns $C' = \mathsf{PRE}^+.\mathsf{ReEnc}(\mathsf{PRE}^+.\mathsf{ReKeyGen}(r', pk_i, pk_j), C_i)$. Derivatives of (pk^*, C^*) are defined as follows:
    1. (pk^*, C^*) is a derivative of itself.
    2. If (pk, C) is a derivative of (pk^*, C^*) and (pk', C') is a derivative of (pk, C), then (pk', C') is a derivative of (pk^*, C^*).
    3. If $\mathcal{A}$ has queried $\mathcal{O}_{re}$ on input (pk, pk', C) and obtained (pk', C'), then (pk', C') is a derivative of (pk, C).
    4. If $\mathcal{A}$ has queried $\mathcal{O}_{rk}$ on input (pk, pk'), and $C' = \mathsf{PRE}^+.\mathsf{ReEnc}(\mathcal{O}_{re}(r', pk, pk'), C)$, then (pk', C') is a derivative of (pk, C).
- $\mathcal{O}_{dec}$: On input (pk_i, C_i), if the following requirements are all satisfied, the Challenger responds as in Phase 1; otherwise, the Challenger outputs $\perp$.
  - pk_i is from $\mathcal{O}_{pk}$;
  - (pk_i, C_i) is not a derivative of (pk^*, C^*).

Guess: Finally, the adversary $\mathcal{A}$ outputs a guess $\mathbf{b}' \in \{0, 1\}$ and wins the game if $\mathbf{b} = \mathbf{b}'$.

We define adversary $\mathcal{A}$'s advantage in attacking PRE^+ as

$$Adv^{2CCA}_{PRE^+} = |\Pr[\mathbf{b} = \mathbf{b}'] - \frac{1}{2}|.$$

Definition 4.3: (CCA-security for the First-level Ciphertext.) *A single-use unidirectional* PRE^+ *scheme is CCA-secure for the first-level ciphertext if the advantage of any PPT adversary* $\mathcal{A}$ *in the following game played between a challenger* $\mathcal{C}$ *and* $\mathcal{A}$ *is negligible in the security parameter* k*. Note that we work in the static corruption model, where the adversary should decide the corrupted users before the game starts.*

Setup: The Challenger sets up the system parameters.

Phase 1: The adversary $\mathcal{A}$ adaptively issues queries $q_1, \cdots, q_{n_1}$ where query q_i is one of:

- $\mathcal{O}_{pk}$: Same as the above definition.
- $\mathcal{O}_{sk}$: Same as the above definition.
- $\mathcal{O}_{rk}$: Same as the above definition.
- $\mathcal{O}_{re}$: Same as the above definition.
- $\mathcal{O}_{dec}$: On input (pk_i, C_i), where pk_i is from $\mathcal{O}_{pk}$, the Challenger returns $\mathsf{PRE}^+.\mathsf{Dec}_1(sk_i, C_i)$, where sk_i is the secret key corresponding to pk_i.

Challenge: Once the adversary $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length plaintexts m_0, m_1 from the message space, and two public keys pk^*, pk on which it wishes to challenge. There are two constraints on the public key pk^*, (i) it is from $\mathcal{O}_{pk}$; (ii) it is uncorrupted. The Challenger picks a random bit $\mathbf{b} \in \{0, 1\}$ and sets

$$C^* = \mathsf{PRE}^+.\mathsf{ReEnc}(\mathsf{PRE}^+.\mathsf{ReKeyGen}(r', pk, pk^*), \mathsf{PRE}^+.\mathsf{Enc}(pk, m_{\mathbf{b}}))$$

It sends C^* as the challenge to $\mathcal{A}$.

Phase 2: The adversary $\mathcal{A}$ adaptively issues more queries $q_{n_1+1}, \cdots, q_n$, but with the following restrictions.

- $\mathcal{O}_{rk}$: On input (pk_i, pk_j) by $\mathcal{A}$, the following requirements should be all satisfied.
  - pk_i and pk_j are from $\mathcal{O}_{pk}$;
- $\mathcal{O}_{dec}$: On input (pk_i, C_i), if the following requirement is satisfied, the Challenger responds as in Phase 1; otherwise, the Challenger outputs $\perp$.
  - pk_i is from $\mathcal{O}_{pk}$;

Guess: Finally, the adversary $\mathcal{A}$ outputs a guess $\mathbf{b}' \in \{0, 1\}$ and wins the game if $\mathbf{b} = \mathbf{b}'$.

We define adversary $\mathcal{A}$'s advantage in attacking PRE^+ as

$$Adv_{PRE^+}^{1CCA} = |\Pr[\mathbf{b} = \mathbf{b}' - \frac{1}{2}|.$$

Remark 1. In the first-level ciphertext security game, the challenge ciphertext is a re-encrypted ciphertext. Note that, in a single-hop unidirectional PRE^+ scheme, since the first level ciphertext cannot be re-encrypted, the adversary should be allowed to obtain any re-encryption keys, even including those from the target public key pk^* to other public keys which are corrupted. Furthermore, since $\mathcal{A}$ is allowed to obtain any re-encryption keys, there is no need to provide the re-encryption oracle and the second level decryption oracle for him.

Definition 4.4: (Collusion Resistant Security.) *A single-use unidirectional* PRE^+ *is collusion resistant under an adaptive chosen ciphertext attack if no polynomial bounded adversary* $\mathcal{A}$ *has a non-negligible advantage against the Challenger in the following game. Here, we also work in the static corruption model.*

Setup: The Challenger sets up the system parameters.

Find: Almost the same as Phase 1 in the above game, except that there is no re-encryption oracle in this game, since the adversary can get every re-encryption key.

Output: Finally, the adversary $\mathcal{A}$ outputs the randomness, which used to generate re-encryption key (According to the above algorithm notation, we can denote it as r'), and wins the game if the corresponding public key pk is uncorrupted.

We define adversary $\mathcal{A}$'s advantage in attacking PRE^+ as

$$Adv^{CR}_{PRE^+} = \Pr[\mathcal{A} \text{ wins}].$$

5 Our Scheme Construction

5.1 The Proposed Single-hop PRE$^+$ Scheme

Our construction is based on [22] and the following is our concrete scheme:

1. $\mathsf{Setup}(k)$: On input the security parameter k, the setup algorithm outputs the system parameter $(g, p, \mathbb{G}, \mathbb{G}_T, e) \rightarrow \mathsf{BSetup}(k)$, hash functions H_1, H_2, H_3 and a random generator $g' \in \mathbb{G}$, where $H_1(\cdot) : \{0,1\}^* \rightarrow Z_p^*, , H_2(\cdot) : \mathbb{G}_T \rightarrow \{0,1\}^l, H_3(\cdot) : \mathbb{G}^2 \times \mathbb{G}_T \times \{0,1\}^l \times \mathbb{G}^4 \rightarrow \mathbb{G}$.

2. $\mathsf{KeyGen}(k)$: On input the security parameter k, the key generation algorithm outputs a (public key, private key) pair $(pk_1 = g^{x_1}, sk_1 = x_1)$ for the decrypter, a (public key, private key) pair $(pk_2 = g^{x_2}, sk_2 = x_2)$ for the delegatee where $x_1, x_2 \in Z_p^*$.

3. $\mathsf{Encrypt}(m, pk_1)$: On input a message $m \in \{0,1\}^l$, fixed randomness (t, r_1) and unfixed randomness (r, k_1, π) from Z_p^*, and a public key pk_1, the regular encryption algorithm outputs $C = (c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9)$ where

$$\begin{aligned} &r = H_1(m, h), c_1 = g^r, c_2 = g^{tr}, c_3 = h \cdot e(g,g)^{rr_1}, \\ &c_4 = m \oplus H_2(h), c_5 = pk_1^{\frac{r_1}{t}}, c_6 = pk_1^{\frac{k_1 r_1}{t}}, c_7 = pk_1^{\pi}, \\ &c_8 = g^{\frac{t}{k_1}}, c_9 = g'^{\frac{k_1 r_1}{t}} H_3(c_1||c_2||c_3||c_4||c_5||c_6||c_7||c_8||pk_1)^{\pi} \end{aligned}$$

   where h is a random element from $\mathbb{G}_T$.

4. $\mathsf{ReKeyGen}(pk_2, \mathsf{Randomness}\ (r_1, t), pk_1)$: The encrypter gives the proxy the re-encryption key

$$rk_{pk_1 \rightarrow pk_2} = (pk_2^{\frac{r_2}{t}}, g^{r_2} g^{-r_1}, g'^{\frac{r_2}{t}}, g^t)$$

   where r_2 is a random number from Z_p^*.

5. $\mathsf{ReEnc}(C, rk_{pk_1 \rightarrow pk2})$: On input the ciphertext C, the proxy outputs $C' = (c'_1, c'_2, c'_3, c'_4, c'_5, c'_6, c'_7, c'_8, c'_9)$:

$$\begin{aligned} &c'_1 = c_1, c'_2 = c_2, c'_3 = c_3 \cdot e(c_1, g^{r_2} g^{-r_1}) = h \cdot e(g,g)^{rr_2}, \\ &c'_4 = c_4, c'_5 = pk_2^{\frac{r_2}{t}}, c'_6 = pk_2^{\frac{k_2 r_2}{t}}, c'_7 = pk_2^{\pi'}, c'_8 = g^{\frac{t}{k_2}}, \\ &c'_9 = g'^{\frac{k_2 r_2}{t}} H_3(c'_1||c'_2||c'_3||c'_4||c'_5||c'_6||c'_7||c'_8||pk_2)^{\pi'} \end{aligned}$$

   where π', k_2. are randomly chosen from Z_p^*.

6. $\mathsf{Decrypt}(C, sk)$: it runs as follows:

   - On input a second-level ciphertext C, and a secret key sk_1, the decryption algorithm is as follows:

– First check whether

$$e(c_9, pk_1) = e(g', c_6)e(H_3(c_1||c_2||c_3||c_4||c_5||c_6||c_7||c_8||pk_1), c_7)$$

if it does not hold, output $\perp$ and terminate, otherwise do the following.

– Compute $h = c_3/e(c_2, c_5)^{\frac{1}{x}}$.

– Compute $m = c_4 \oplus H_2(h)$.

– Compute $r = H_1(m, h)$.

– Output m, if $c_1 = g^r$, $\perp$ otherwise.

• On input a first-level ciphertext C', and a secret key sk_2, the decryption algorithm decrypts as the above, for these two level ciphertexts share the same structure.

Remark 2. Roughly speaking, this scheme can be viewed as combining two parts: the BF-IBE part and the Waters signature part. Note that c_4 is computed by the scheme BF [4], and c_6, c_7, c_9 are computed by the strongly unforgeable variant [6] of Waters scheme [42]. Thus our scheme's security roughly can be based on the security of BF-IBE and Waters signature.

Remark 3. By combing randomness (r, r_1, π) and the introduced randomness (k, t), we can instantiate the concept of PRE^+ and prove its security. It should be noted that our scheme can be simplified or more natural schemes in the standard model can be constructed, we leave them as important open problems.

5.2 Security Analysis

Theorem 5.1: *Our proposal is PRE$^+$-CCA-secure in the random oracle model for the second level ciphertext under assumptions that the eDBDH problem and eDL problem are hard.*

Proof. We incrementally define a sequence of games starting at the real attack (**Game** G_0), and ending up at **Game** G_9, which clearly shows that the adversary cannot break the system. We define E_i to be the event $b = b'$ in Game G_i, where b is the bit involved in the challenge phase, and b' is the output of $\mathcal{A}$ in the Guess phase. Note in our scheme, the two-level ciphertexts share the same structure, thus the decryption oracle for the first level ciphertext and the decryption oracle the second level ciphertext are same, thus in the following Games, we just consider one decryption oracle.

• **Game** G_0**.** This game corresponds to the real attack. By definition,

$$|Pr[E_0] - 1/2| = Adv^{2CCA}_{PRE^+}(k) \tag{1}$$

• **Game** G_1**.** In this game, we modify the hash functions to the random oracles.

– O_{h_1}: On input a string S, the Challenger checks whether $(S, r^{(1)})$ exists in Table T_{h_1}. If it is, the Challenger returns $r^{(1)}$; otherwise, the challenger chooses a random number $r^{(1)}$ from Z_p^*, returns $r^{(1)}$, and records $(S, r^{(1)})$ into Table T_{h_1}.

– O_{h_2}: On input a string S, the Challenger checks whether $(S, R^{(2)})$ exists in Table T_{h_2}. If it is, the Challenger returns $R^{(2)}$; otherwise, the Challenger chooses a random number $R^{(2)}$ from $\{0,1\}^l$, returns $R^{(2)}$, and records $(S, R^{(2)})$ into Table T_{h_2}.

– O_{h_3}: On input a string $c_1||c_2||c_3||c_4||c_5||||c_6||c_7||c_8||pk$, the Challenger checks whether $(c_1||c_2||c_3||c_4||c_5||||c_6||c_7||c_8||pk, R^{(3)}, *)$ exists in Table T_{h_3}. If it is, the Challenger returns $R^{(3)}$; otherwise, the Challenger chooses a random number $r^{(3)}$ from Z_p^*, returns $R^{(3)} = pk^{r^{(3)}}$, and records $(c_1||c_2||c_3||c_4||c_5||||c_6||c_7||c_8||pk, R^{(3)}, r^{(3)})$ in Table T_{h_3}.

• **Game** G_2**.** In this game, we modify O_{h_1} and the challenge phase by setting $h = h^*$, where h is a random number from Z_p^*.

– O_{h_1}: On input a string $m_b||h^*$, the Challenger aborts. Other performances are the same as those in O_{h_1} which we first mention.

– Challenge: The Challenger sets $h = h^*$, other performances are the same as those in O_{h_1} which we first mention.

It is easy to see that only if the adversary never queries O_{h_1} with $m_b||h^*$, **Game** G_2 and **Game** G_1 are indistinguishable. Hence, we have

$$|Pr[E_2 - E_1] \leq \frac{q_{h_1}}{p}$$

where q_{h_1} is the maximum number of queries to O_{h_1} (including the number the Challenger queries O_{h_1}).

- **Game** G_3**.** In this game, we set $g' = g_{+1}(\neq g) = g^u$, where u is chosen randomly from Z_p^*. The Challenger knows a value $log_g^{g_{-1}} = 1/u$ satisfying $e(g_{-1}, g_{+1}) = e(g, g)$. It is easy to see that **Game** G_3 and **Game** G_2 are indistinguishable.

$$Pr[E_3] = Pr[E_2]$$

- **Game** G_4**.** In this game, we modify the re-encryption oracle and the decryption oracle as follows.

  – O_{re}: On input (pk_i, pk_j, C), if $(pk^*, C^*) = (pk_i, C)$ and pk_j is uncorrupted, the Challenger records $(pk_j, Encrypt(pk_j, O_{dec}(pk_i, C)))$ into Table T_{re}. Other performances are the same as that in Game G_3.

  – O_{dec}: On input (pk_i, C_i), if pk_i is uncorrupted, the Challenger checks whether (pk_i, C_i) exists in Table T_{re}. If yes, the Challenger outputs $\perp$. Other performances are the same as that in Game G_4. Note that Table T_{re} is used to record the derivatives of the second level challenge ciphertext from O_{re}.

  we can easily obtain that

$$Pr[E_4] = Pr[E_3]$$

- **Game** G_5**.** In this game, we modify the decryption oracle as follows.

  – O_{dec}: The Challenger only changes the method of doing the decryption for the uncorrupted public keys as follows, and other performances are the same as that in Game G_4.

  1. Set a set $\mathcal{S} = \phi$.
  2. Find $(S, r^{(1)})$ in Table T_{h_1}, such that $g^{r^{(1)}} = c_1, m'||h' = S, c_4 = m' \oplus H_2(h')$.
  3. If $\mathcal{S} = \phi$, then output $\perp$ and terminate. Otherwise, do the following.
  4. Choose the first pair $(S, r^{(1)})$ from $\mathcal{S}$, and set $\mathcal{S} = \mathcal{S}/(S, r^{(1)})$.
  5. If $h' = c_3/e(c_9/c_7^{r^{(3)}}, c_8)^{r^{(1)}/u}$ and $e(c_9, pk_i) = e(g', c_6)e(H_3(c_1||c_2||c_3||c_4||c_5||c_6||c_7||c_8||pk_i), c_7)$ both hold, output m' and terminate; otherwise, go to Step 3. Note that $r^{(3)}$ is the associated value to $(c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, pk_i)$ in T_{h_3}, and

$$H_3(c_1||c_2||c_3||c_4||c_5||c_6||c_7||c_8||pk_i) = pk_i^{r^{(3)}}$$

In Game G_5, the Challenger decrypts the ciphertext not by using the secret key but the random oracles (O_{h_1}, O_{h_2} and O_{h_3}). Hence, Game G_5 and Game G_4 would be different, if the adversary guessed the correct value of $H_1(m||h)$ without querying O_{h_1} or guessed the correct value of $H_2(h)$ without querying O_{h_2} or guessed the correct value of $H_3(c_1||c_2||c_3||c_4||c_5||c_6||c_7||$
$c_8||pk_i)$ without querying O_{h_3}. The probability of this issue is $(q_{h_2}/p + q_{h_3}/2^l + q_{h_3}/p)$ at most. As a result, we have that

$$|Pr[E_5] - Pr[E_4]| \leq q_{dec}(q_{h_2}/p + q_{h_3}/2^l + q_{h_3}/p)$$

where $q_{h_1}, q_{h_2}, q_{h_3}$ is the maximum number of queries to $O_{h_1}, O_{h_2}, O_{h_3}$ (including the number the Challenger queries $O_{h_1}, O_{h_2}, O_{h_3}$).

- **Game** G_6**.** In this game, we continue to modify the decryption oracle as follows.

– It is almost the same as that in Game G_5, except that if pk_i is corrupted, and the plaintext is $\perp$ obtained in the previous performances, the Challenger additionally performs the following.

1. If $e(c_9, pk_i) = e(g', c_6)e(H_3(c_1||c_2||c_3||c_4||c_5||c_6||c_7||c_8||pk_i), c_7)$ holds, do the following steps; otherwise, output $\perp$ and terminate.
2. Set $\mathcal{S}_1 = \mathcal{S}_2 = \mathcal{S}_2'$.
3. Search $(rk^1_{pk^*\to pk_i}, rk^2_{pk^*\to pk_i}, rk^3_{pk^*\to pk_i}, rk^4_{pk^*\to pk_i})$ in Table T_{rk} (where T_{rk} records the re-encryption keys which have returned to the adversary), and put all $(rk^1_{pk^*\to pk_i}, rk^2_{pk^*\to pk_i}, rk^3_{pk^*\to pk_i}, rk^4_{pk^*\to pk_i})$ into the set $\mathcal{S}_1$.
4. Search $(S, r^{(1)})$ in Table T_{h_1}, such that $m'||h' = S$, $c_4 = m' \oplus H_2(h')$, and $c_1 = g^{r^{(1)}}$, and then put all $(S, r^{(1)})$ into set $\mathcal{S}_2$ and $\mathcal{S}_2'$.
5. If $\mathcal{S}_1 = \phi$ or $\mathcal{S}_2 = \phi$, then output $\perp$ and terminate. Otherwise, do the following.
6. Choose the first pair $(rk^1_{pk^*\to pk_i}, rk^2_{pk^*\to pk_i}, rk^3_{pk^*\to pk_i}, rk^4_{pk^*\to pk_i})$ from $\mathcal{S}_1$, and set $\mathcal{S}_1 = \mathcal{S}_1/(rk^1_{pk^*\to pk_i}, rk^2_{pk^*\to pk_i}, rk^3_{pk^*\to pk_i}, rk^4_{pk^*\to pk_i})$
7. Choose the first pair $(S, r^{(1)})$ from $\mathcal{S}_2$, and set $\mathcal{S}_2 = \mathcal{S}_2/(S, r^{(1)})$.
8. If $c_3' = c_3 e(c_9/c_7^{r^{(3)}}, g^{\frac{t}{k_1}})^{\frac{r^{(1)}}{u}}$ where $r^{(3)}$ is the associated value to $(c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, pk_i)$ in T_{h_3}.
9. If $\mathcal{S}_1 = \phi$, output $\perp$; otherwise, do the following steps.
10. If $\mathcal{S}_2 = \phi$, then set $\mathcal{S}_2 = \mathcal{S}_2'$ and go to Step 5; otherwise, go to Step 6.

Note that if the input regular ciphertext is computed from ReEnc, the verifying equation in step 8 must hold. Hence, the modification in this game does not make Game G_6 different from Game G_5. As a result, we have that

$$Pr[E_6] = Pr[E_5]$$

- **Game** G_7. In this game, we continue to modify the re-encryption key generation oracle as follows.

  – O_{rk}: On input (pk_i, pk_j), if $pk_i \neq pk^*$, then the Challenger returns $(pk_j^{r_2/t}, g^{r_2}g^{-r_1}, g'^{r_2/t}, g^t)$ where (r_1, t) are fixed randomness chosen by the challenger in advance, and r_2 is randomly chosen from Z_p^*. Other performances are the same as those in Game G_6.

It is easy to see that Game G_7 and Game G_6 are indistinguishable. Hence, we have that

$$Pr[E_7] = Pr[E_6]$$

- **Game** G_8. In this game, we modify the re-encryption oracle as follows.

  – O_{rk}: On input (pk_i, pk_j), if $pk_i = pk^*$, the pk_j is uncorrupted, then the Challenger returns (A, B, C, D, pk_j) where A, B, C are random elements from G, and $e(A, g') = e(C, pk_j)$ and D is fixed as g^t. Other performances are the same as those in Game G_7.

We have the following analysis on the difference between Game G_7 and Game G_8.

– O_{dec} and O_{re} do not use the secret keys of the uncorrupted users in Game G_7.

Note that due to the modification in Game G_6, the decryption oracle cannot help to verify the modification on O_{rk}. In Game G_6, it is guaranteed that the first level ciphertexts from ReEnc(O_{rk}, C) can be decrypted properly.

Hence, if the adversary without knowing $1/u$ can verify the modification on O_{rk}, we can build an algorithm to solve the eDL hard problem. In particular, set $t = 1/\alpha_1, pk_i = g^{\alpha_2}, r_1 = \alpha_4, pk_i^{r_1/t} = g^{\alpha_1\alpha_2\alpha_4}, pk_j = g^{\alpha_3}, r_2 = \alpha_5, (pk_j)^{r_2/t} = g^{\alpha_1\alpha_3\alpha_5}, (g')^{r_2/t} = g^{\alpha_1\alpha_4 u}$, we can not calculate the value of $g^{r_2}g^{-r_1} = g^{\alpha_5-\alpha_4}$. Here we do not consider the information about (r_2, r_1) leaked from $(c_2, c_3, c_6, c_8, c_9, c_2', c_3', c_6', c_8', c_9')$ for

these value all are completely random for randomness $(r,(r,h),k_1,(k_1,\pi),r,(r,h),k_2,(k_2,\pi'))$. Hence we have that

$$|Pr[E_8] - Pr[E_7]| \leq q_{rk} \cdot \epsilon_{eDL}$$

where q_{rk} is the maximum number of queries to O_{rk}.

- **Game** G_9. In this game, we modify the challenge phase as follows.
  - Challenge: The challenge ciphertext is

$$r = a, c_1^* = g^r = g^a, t = b, c_2^* = g^{tr} = g^{ab}, c_3^* = h^* T, c_4^* = m_b \oplus H_2(h^*), pk_1 = g^c,$$

$$r_1 = d, c_5^* = pk_1^{r_1/t} = g^{cd/b}, k_1 = e, c_6^* = pk_1^{k_1 r_1/t} = g^{cde/b}, \pi = f, c_7 = pk_1^{\pi} = g^{cf},$$

$$c_8 = g^{t/k_1} = g^{b/e}, H_3(c_1||c_2||c_3||c_4||c_5||c_6||c_7||c_8||pk_1) = g^s,$$

$$c_9 = g'^{\frac{k_1 r_1}{t}} H_3(c_1||c_2||c_3||c_4||c_5||c_6||c_7||c_8||pk_1)^{\pi} = g^{ued/b} g^{sf}$$

The rest of performances are the same as those in previous game. Note that in the above we implicitly set $H_1(m,h) = a$, and T is a random number. If the adversary can distinguish Game G_9 and Game G_8, we can build an algorithm B that solves the eDBDH problem. The eDBDH problem is, given $(g^a, g^{ab}, g^{cd/b}, g^{cde/b}, g^{cf}, g^{b/e}, g^{ued/b} g^{sf}, T)$ and decide if $T = e(g,g)^{ad}$ or random. it is clear that Game G_9 is the same as Game G_8. Hence, we have that

$$|Pr[E_9] - Pr[E_8]| \leq \epsilon_{eDBDH}$$

Furthermore, it is clear that in Game G_9

$$Pr[E_9] = 1/2$$

since T is a random element.

As a result, combining the above equations among the Games, we get the theorem proved.

Theorem 5.2: *Our proposal is PRE$^+$-CCA-secure in the random oracle model for the first level ciphertext under assumptions that the eDBDH problem and eDL problem are hard.*

We can use a similar method in the proof of the above Theorem 5.1 to prove this theorem, for the first level ciphertext and the second level ciphertext share the same structure, here we omit the details..

Theorem 5.3: *Our PRE$^+$ proposal is collusion-resistent under assumption that vDL problem is hard.*

Here when the proxy and the delegatee collude, they can compute $(g^{\frac{r_2}{t}}, g^{r_2} g^{-r_1}, g'^{\frac{r_2}{t}}, g^t)$, but this does not help them compute (r_1, t) under the eDL problem, which is the randomness used to generate re-encryption keys. The vDL problem is, given $(g^{\beta_1}, g^{u\beta_2/\beta_1}, g^{\beta_2 - \beta_3}, g^{\beta_2/\beta_1})$, compute (β_2, β_1), which is difficult. Thus our PRE$^+$ proposal is collusion-resistent.

5.3 Feature Comparison and Performance Analysis

In this subsection, we first give the feature comparison results with related work, which can be seen in Table 1. From this table, we can see that our scheme can achieve most useful properties for cloud storage, such as being unidirectional, CCA-secure, supporting conditional delegation, collusion-safe, without cost bilinear pairing, non-transferable and fine-grained message-level delegation. Especially, for the last two properties, non-transferability and message-level delegation, until now almost no natural proxy re-encryption schemes could be achieved.

Further, we give the performance analysis results, which can be seen in Table 2. Here we just compare with scheme [22], on which we construct our scheme. Here t_m denotes the computation time for modular exponentiation and t_p denotes the computation time for bilinear pairing, $|G|$ denotes the length of one group element in $|G|$, $|G_T|$ denotes the length of one group element in G_T, l denotes the length of the message or the length of the hash function

Table 1 Feature Comparison

Schemes	Unidirectional	(R)CCA-secure	Conditional	Collusion-safe	Without pairing	Non-transferable	Message-level delegation
[1]	Yes	No	No	Yes	No	No	No
[8]	No	Yes	No	No	No	No	No
[13]	No	Yes	Yes	No	Yes	No	No
[22]	Yes	Yes	Yes	Yes	Yes	No	No
[31]	Yes	Yes	No	Yes	No	No	No
[46]	Yes	Yes	Yes	Yes	No	No	No
Our	Yes	Yes	Yes	Yes	No	Yes	Yes

H_2's output (our scheme can be easily modified to handle arbitrary length message). From this table we can see that our scheme is a little less efficient than scheme of [22], but our scheme can achieve many other interesting properties, while their scheme can not achieve them. Furthermore, we believe that there can be far more efficient PRE^+ schemes than this scheme, which is just a proof of concept, we leave designing such schemes as our future work.

Table 2 Computation cost

Scheme	Enc	Check	Reenc	Dec		Ciph-Len	
				1ndCiph	2ndCiph	1stCiph	2ndCiph
[22]	$2t_m + 1t_p$	$2t_p$	$5t_m + 1t_p$	$2t_p$	$4t_p + 3t_m$	$4\|G\| + 1\|G_T\| + \|l\|$	$3\|G\| + \|l\|$
Our	$1t_p + 8t_e$	$3t_p$	$5t_m + 1t_p$	$2t_m + 4t_p$	$2t_m + 4t_p$	$7\|G\| + 1\|G_T\| + \|l\|$	$7\|G\| + 1\|G_T\| + \|l\|$

5.4 Extention to a Multi-hop Variant

Our single-hop PRE^+ scheme can be easily extended to a multi-hop variant. Below is the concrete scheme:

1. $\mathsf{Setup}(1^k)$: Same as the single-hop scheme.
2. $\mathsf{Keygen}(1^k)$: Same as the single-hop scheme.
3. $\mathsf{Encrypt}(m, pk_{i-1})$: Same as the single-hop scheme.
4. $\mathsf{ReKeyGen}(pk_i, \mathsf{Randomness}\ (r_{i-1}, t), pk_{i-1})$: The encrypter gives the proxy the re-encryption key

$$rk_{pk_i} = (pk_i^{\frac{r_i}{t}}, g^{r_i} g^{-r_{i-1}}, g'^{\frac{r_i}{t}}, g^t)$$

   where r_i is a random number from Z_p^*, and the encrypter records r_i for the next hop's re-encryption key generation.

5. $\mathsf{ReEnc}(C, rk_{pk_{i-1} \to pk_i})$: Same as the single-hop scheme.
6. $\mathsf{Decrypt}(C, sk_i)$: Same as the single-hop scheme.

Remark 4. Note in this scheme the ciphertext remains constant no matter how many times it has been re-encrypted. This is a very desirable feature in practical applications. We emphasise the construction of constant-size ciphertext multi-hop CCA-secure traditional PRE scheme remains as a very important open problem until now. We also note the encrypter needs to remember the randomness r_i for the $(i+1)$ hop's re-encryption key generation, which will restrict its employment. We leave construction of new multi-hop PRE^+ scheme, which need not remember the randomness as an important open problem.

6 Scanf: A scalable and controllable secure cloud data sharing framework

In this section, we describe a scalable and controllable secure cloud data sharing framework, denoted as *Scanf*, based on the primitive PRE$^+$, which can be seen in Fig. 4.

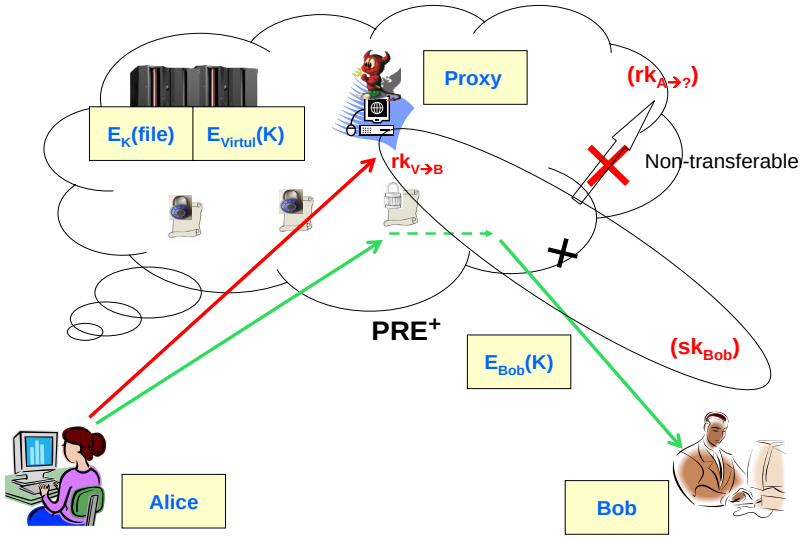

Figure 4 PRE$^+$ in Secure Cloud Computation

In this framework, there are four kinds of parties. The first kind is the client users, denoted as Alice, which outsources data contents to the Cloud. The second one is the infrastructure provider, the Cloud, which stores and manages client users' outsouring data contents. The third one is the data users (sharers), denoted as Bob, which will access Alice's data contents. The final one is the access controller, denoted as proxy, which does the ciphertext transformation process (note here the proxy can be the Cloud). In order to protect her outsourced data's privacy, Alice first encrypts her data contents. Here we assume Alice first constructs a virtual user Virtual, which she knows its private/public key. Then Alice encrypts her data content under a symmetrical key K, and then encrypts K under the public key of Virtual, and outsources them to Cloud. After knowing Bob's public key, she assigns the re-encryption key $rk_{v\to B}$ to the proxy. When data user Bob wants to access the content of Alice, he relies on the proxy doing the re-encryption from V to B with $rk_{v\to B}$. In this process, the proxy cannot know the content, neither he can get any information on private keys of Virtual and Bob.

One might argue that the traditional PRE can also do this job. But our point is that our Scanf framework based on PRE$^+$ performs very well on the **non-transferable and message level based fine-grained delegation issues**. In the current state of the art, almost there exist no PRE schemes with non-transferable property, and no IBPRE schemes with non-transferable property without the help of PKG [44], while relying on PKG is not a good choice for practical applications. That means, if using PRE in this framework, the proxy and Bob can collude to representing Alice, to make new delegations to new users. Obviously, this contradicts Alice's original intention. Similarly, in the current state, all the PRE schemes cannot handle the fine-grained delegation at the **message based level**. This will be inconvenient as Alice has no control on which content will be re-encrypted. She has to incorporate the condition or type or time information in her content if she uses the conditional or type or time based PRE. While in PRE$^+$, Alice can easily control on which content will be re-encrypted and which will be not. Thus our framework can achieve good scalability and controllability, which is important for secure data sharing in cloud storage.

7 Conclusion

In this paper, we have introduced a new primitive: PRE^+, which is the dual of PRE. We have discussed the differences with existing primitives PRE and BE. We also give the definition and a security model for this new primitive, construct a concrete single-hop scheme and prove its IND-CCA security, and finally,extend it to a multi-hop scheme. Based on our scheme, we proposed a scalable and controllable secure cloud data sharing framework, denoted Scanf, which is useful for practical applications of data storage.

In our future work we would like to explore other issues such as finding interesting applications of PRE^+, constructing other variants of single-hop and multi-hop PRE^+ schemes based on different assumptions, exploring the relationship between PRE^+ and other cryptographic primitives like proxy encryption, etc.

8 Acknowledgements

The authors would like to express their gratitude thanks to Dr. Xiaoqi Jia, Dr. Qianhong Wu, Dr. Jian Weng, Dr. Jun Shao, Dr. Licheng Wang for many helpful comments. This work was supported by Natural Science Foundation of Shaanxi Province (Grant No. 2014JM8300), the Changjiang Scholars and Innovation Research Team in University (Grant NO. IRT 1078), the Key Problem of NFSC-Guangdong Union Foundation (Grant NO. U1135002), the Major Nature Science Foundation of China (Grant NO. 61370078), China 863 project, the Fundamental Research Funds for the Center Universities (Grant NO. JY10000903001), Nature Science Foundation of China (Grant NO. 61103230, 61272492, 61402531).

Fatos Xhafa's work has been supported by the Spanish Ministry for Economy and Competitiveness (MINECO) and the European Union (FEDER funds) under grant COMMAS (ref. TIN2013-46181-C2-1-R).

References

[1] G. Ateniese, K. Fu, M. Green and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *ACM NDSS 2005*, pages 29–43, 2005.

[2] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *ACM Transactions on Information and System Security*, no. 1, pages 1–30. 2006.

[3] G. Ateniese, K. Benson, and S. Hohenberger. Key private proxy re-encryption. In *CT-RSA 2009*, volume 5473 of *LNCS*, pages 279–294, 2009.

[4] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229, 2001.

[5] D. Boneh, C. Gentry and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275, 2005.

[6] D. Boneh, E. Shen, and B. Waters. Strongly unforgeable signatures based on computational Diffie-Hellman. In *PKC 2006*, volume 3985 of *LNCS*, pages 229-240, 2006.

[7] M. Blaze, G. Bleumer and M. Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT 1998*, volume 1403 of *LNCS*, pages 127–144, 1998.

[8] R. Canetti and S. Hohenberger. Chosen ciphertext secure proxy re-encryption. In *ACM CCS 2007*, pages 185–194, 2007.

[9] R. Canetti, S. Halevi and J. Katz. A forward-secure public-key encryption scheme. In *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 255–271, 2003.

[10] Y-P.Chiu, C-L. Lei, and C-Y. Huang. Secure multicast using proxy encryption. In *ICICS 2005*, volume 3783 of LNCS, pages 280–290, 2005.

[11] C. Chu and W. Tzeng. Identity-based proxy re-encryption without random oracles. In *ISC 2007*, volume 4779 of *LNCS*, pages 189–202, 2007.

[12] C. Chu, S. Chow, J.Weng, J. Zhou, and R. H. Deng. Conditional proxy broadcast re-encryption. In *ACISP 2009*, volume 5594 of *LNCS*, pages 327–342, 2009.

[13] R. Deng, J. Weng, S. Liu and K. Chen. Chosen ciphertext secure proxy re-encryption without pairing. In *CANS 2008*, volume 5339 of *LNCS*, pages 1–17, 2008.

[14] A. Ivan and Y. Dodis. Proxy cryptography revisited. In *Internet Society (ISOC): NDSS 2003*, 2003.

[15] L. Fang, W. Susilo and J. Wang. Anonymous conditional proxy re-encryption without random oracles. In *PROVSEC 2009*, volume 5854 of *LNCS*, pages 47–60, 2009.

[16] M. Green and G. Ateniese. Identity-based proxy re-encryption. In *ACNS 2007*, volume 4521 of *LNCS*, pages 288–306, 2007.

[17] T. S. Heydt-Benjamin, H. Chae, B.Defend, and K. Fu. Privacy for public transportation. In *PET 2006*, volume 4258 of LNCS, pages 1–19, 2005.

[18] Y. He, T. Chim, L. Hui and S. Yiu. Non-Transferable Proxy Re-Encryption Scheme for Data Dissemination Control. Available at Cryptology ePrint Archive, Report 2010/192, 2010.

[19] S. Hohenberger, G. N. Rothblum, A.Shelat, and V. Vaikuntanathan. Securely obfuscating re-encryption. In *TCC 2007*, volume 4392 of LNCS, pages 233–252, 2007.

[20] S. Hohenberger, B. Waters. Realizing hash-and-sign signatures under standard assumptions. In *EUROCRYPT 2009*, volume 5479 of LNCS, pages 333–350, 2009.

[21] L. Ibraimi, Q. Tang, P. Hartel, and W. Jonker. A type-and-identity-based proxy re-encryption scheme and its application in healthcare. In *SDM 2008*, volume 5159 of LNCS, pages 185–198, 2008.

[22] X. Jia, J. Shao, J. Jing and P. Liu. CCA-secure type-based proxy re-encryption with invisible proxy. In *2010 10th IEEE International Conference on Computer and Information Technology (CIT 2010)*, pages 1299–1305.

[23] H. Khurana, A. Slagell, and R.Bonilla. Sels: A secure e-mail list service. In *ACM SAC 2005*, pages 306-313, 2005.

[24] H. Khurana and H-S.Hahm. Certified mailing lists. In *ASIACCS 2006*, pages 46–58, 2006.

[25] H. Khurana and R. Koleva. Scalable security and accounting services for content-based publish subscribe systems. In *International Journal of E-Business Research*, 2(3), 2006.

[26] K. Kurosawa and Y. Desmedt. A new paradigm of hybrid encryption scheme. In *CRYPTO 2004*, volume 3152 of *LNCS*, pages 426–442, 2004.

[27] K. Liang, W. Susilo, J. K. Liu. Privacy-Preserving Ciphertext Multi-Sharing Control for Big Data Storage. In *IEEE Transactions on Information Forensics and Security*, Vol. 10, No. 8, pages 1578–1589, 2015.

[28] K. Liang, M. H. Au, J. K. Liu, W. Susilo, D. S. Wong, G. Yang, T. Phuong, Q. Xie. A DFA-Based Functional Proxy Re-Encryption Scheme for Secure Public Cloud Data Sharing. In *IEEE Transactions on Information Forensics and Security*, Vol. 9, No. 10, pages 1667–1680, 2014.

[29] X. Liang, Z. Cao, H. Lin, and J. Shao. Attribute based proxy re-encryption with delegating capabilities. In *AISACCS 2009*,pages 276–286, 2009.

[30] B. Libert and D. Vergnaud. Tracing malicious proxies in proxy re-encryption. In *Pairing 2008*, volume 5209 of *LNCS*, pages 332–353, 2008.

[31] B. Libert and D. Vergnaud. Unidirectional chosen ciphertext secure proxy re-encryption. In *PKC 2008*, volume 4939 of *LNCS*, pages 360–379, 2008.

[32] T. Matsuo. Proxy re-encryption systems for identity-based encryption. In *Pairing 2007*, volume 4575 of *LNCS*, pages 247–267, 2007.

[33] J. Shao and Z. Cao. CCA-secure proxy re-encryption without pairing. In *PKC 2009,* LNCS 5443, pages. 357–376, Springer–Verlag, 2009.

[34] J. Shao, Z. Cao, P. Lin. Generic construction for CCA-secure unidirectional proxy re-encryption. In Security and Communication Networks, no. 2, pages 1-16, 2009.

[35] J. Shao, Z. Cao and P. Lin. CCA-secure PRE scheme without random oracle. Cryptology ePrint Archive, Report 2010/112, 2010. Available at http://eprint.iacr.org.

[36] J. Shao, Z. Cao, X. Liang and H. Lin. Proxy re-encryption with keyword search. In *Information Science 2010*, Doi: 10.1016/j.ins.2010.03.026, 2010.

[37] J. Shao, R. Lu, X. Lin. Fine-grained data sharing in cloud computing for mobile devices. In *INFOCOM 2015*, 2677-2685, 2015.

[38] J. Shao, R. Lu, X. Lin. FINE: A fine-grained privacy-preserving location-based service framework for mobile devices. In *INFOCOM 2014*, 244-252, 2014.

[39] Q. Tang. Type-based proxy re-encryption and its construction. In *INDOCRYPT 2008*, volume 5365 of *LNCS*, pages 130–144, 2008.

[40] G. Taban, A. A. Cardenas, and V. D. Gligor. Towards a secure and interoperable drm architecture. In *ACM DRM 2006*, pages 69–78, 2006.

[41] A. Talmy and O. Dobzinski. Abuse freedom in access control schemes. In *AINA 2006*, pages 77–86, 2006.

[42] B. Waters. Efficient identity-based encryption without random oracles. In *Eurocrypt 2005*, volume 3494 of LNCS, pages 114–127, 2005.

[43] L. Wang, L. Wang, M. Mambo, E. Okamoto. New identity-based proxy re-encryption schemes to prevent collusion attacks. In *Pairing 2010*, volume 6487 of LNCS, pages 327–346, 2010.

[44] X. Wang, X. Yang, F. Li. On the Role of PKG for Proxy Re-encryption in the Identity Based Setting. Available at Cryptology ePrint Archive, Report 2008/410, 2008.

[45] J. Weng, R. H. Deng, C. Chu, X. Ding, and J. Lai. Conditional proxy re-encryption secure against chosen-ciphertext attack. In *ACM ASIACCS 2009*, pages 322–332, 2009.

[46] J. Weng, Y. Yang, Q. Tang, R. Deng, and F. Bao. Efficient conditional proxy re-encryption with chosen-ciphertext security. In *ISC 2009*, volume 5735 of *LNCS*, pages 151–166, 2008.

[47] J. Weng, M. Chen, Y. Yang, R. Deng, K. Chen and F. Bao CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles. In Science China Information Sciences, 53:593-606, 2010.

[48] J. Weng, M. Chen, Y. Yang, R. Deng, K. Chen and F. Bao CCA-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles. Cryptology ePrint Archive, Report 2010/265, 2010. Available at http://eprint.iacr.org.

[49] S. Chow, J. Weng, Y. Yang, R. Deng Efficient unidirectional proxy re-encryption. In *AFRICACRYPT 2010*, volume 6055 of *LNCS*, pages 316–332, 2010.