This is the peer reviewed author accepted manuscript (post print) version of a published work that appeared in final form in:

Yang, Yanjiang, Zhu, Haiyan, Lu, Haibing, Weng, Jian, Zhang, Youcheng, Choo, Kim-Kwang Raymond 2016 'Cloud based data sharing with fine-grained proxy re-encryption', Pervasive and Mobile Computing, vol. 28, pp. 122-134

This output may not exactly replicate the final published authoritative version for which the publisher owns copyright. It is not the copy of record. This output may be used for non-commercial purposes. This version is licensed under a Creative Commons CC-BY-NC-ND 4.0 license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

The final definitive published version (version of record) is available at: https://doi.org/10.1016/j.pmcj.2015.06.017

Persistent link to the UniSA Research Outputs Repository: https://researchoutputs.unisa.edu.au/11541.2/118802

General Rights

Copyright and moral rights for the publications made accessible in the UniSA Research Outputs Repository are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognize and abide by the legal requirements associated with these rights.

- Users may download and print one copy for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the persistent link identifying the publication in the UniSA Research Outputs Repository

If you believe that this document breaches copyright, please contact us and provide details. We will remove access to the work immediately and investigate your claim.

Version of Record: https://www.sciencedirect.com/science/article/pii/S1574119215001248
Manuscript_f297feffa6baf471356b3bf188565616

Click here to view linked References

Cloud Based Data Sharing with Fine-grained Proxy Re-encryption

Yanjiang Yang[a,*], Haiyan Zhu[b], Haibing Lu[c], Jian Weng[d], Youcheng Zhang[e], Kim-Kwang Raymond Choo[f]

[a] *Institute for Infocomm Research, Singapore*
[b] *School of Electronic and Information Engineering, Beihang University, China*
[c] *The Leavey School of Business, Santa Clara University, USA*
[d] *Department of Computer Science, Jinan University, China*
[e] *Nanjing Unary Information Technology Co., Ltd, China*
[f] *Information Assurance Research Lab, University of South Australia, Australia*

Abstract

Conditional proxy re-encryption (CPRE) enables fine-grained delegation of decryption rights, and has many real-world applications. In this paper, we present a ciphertext-policy attribute based CPRE scheme, together with a formalization of the primitive and its security analysis. We demonstrate the utility of the scheme in a cloud deployment, which achieves fine-grained data sharing. This application implements cloud server-enabled user revocation, offering an alternative yet more efficient solution to the user revocation problem in the context of fine-grained encryption of cloud data. High user-side efficiency is another prominent feature of the application, which makes it possible for users to use resource constrained devices, e.g., mobile phones, to access cloud data. Our evaluations show promising results on the performance of the proposed scheme.

Keywords: Cloud computing, Conditional proxy re-encryption, User revocation, Fine-grained encryption

1. Introduction

The notion of proxy re-encryption (PRE) was first introduced by Blaze, Bleumer and Strauss [1]. In a PRE scheme, a semi-trusted proxy is given a re-encryption key and, thus, is able to convert ciphertexts under Alice's public key into ciphertexts under Bob's public key. The proxy, however, is unable to learn any useful information about the messages encrypted under either key. This re-encryption procedure can be intuitively depicted as $E(pk_A, \cdot) \xrightarrow{rk_{A\to B}} E(pk_B, \cdot)$, where $rk_{A\to B}$ denotes the re-encryption key from Alice to

*Corresponding author, Email: yyang@i2r.a-star.edu.sg; Tel: (65)98565982

© 2015 published by Elsevier. This manuscript is made available under the Elsevier user license
https://www.elsevier.com/open-access/userlicense/1.0/

Bob, and $E(pk, \cdot)$ denotes a ciphertext under public key pk[1]. In this setting, Alice is the *delegator* and Bob is the *delegatee*, and PRE is said to enable Alice to delegate her decryption right to Bob.

A weakness of traditional PRE is that the proxy can transform *all* of Alice's ciphertexts, disallowing Alice to choose a subset of her ciphertexts to be converted. This is not satisfactory in many applications where fine-grained delegations of decryption right are required. This has led to the emergence of the concept of *conditional* proxy re-encryption (CPRE) [2, 3], which strengthens PRE such that transformation of ciphertexts by the proxy is conditional: a ciphertext under Alice's public key is generated by embedding in the ciphertext a value w, and the re-encryption key from Alice to Bob is associated with another value w'; the ciphertext can be converted to a ciphertext for Bob if and only if $w = w'$. Intuitively, the procedure is depicted as $E(pk_A, \cdot, w) \xrightarrow{rk_{A \xrightarrow{w'} B}} E(pk_B, \cdot)$. CPRE turns out to be a useful primitive, and can be used in many real-world applications. In this paper, we show one such example: applying CPRE to achieve *revocable* fine-grained data sharing over cloud storage services.

Ensuring the security and privacy of data stored in external cloud storage services is of paramount importance to organizational and individual users, and one typical means of achieving this is using encryption (by the data owner) [4, 5, 6]. However, encryption of cloud data poses various challenges. First of all, cloud storage services are an effective platform to share data among a number of stakeholders and users. In practice, fine-grained access control policies are often deployed to ensure that different users have different access privileges. This leads to the first challenge that encryption of cloud data must support fine-grained cryptographic access control such that users with different privileges possess different decryption capabilities. The second challenge for data sharing in the multi-user setting is user revocation, where one or more users would need to be deprived of their access rights in certain circumstances, e.g., they resign from the organization. The typical solution to user revocation is to invalidate the existing key (used in data encryption) by re-generating the encrypted data with a new key, which is then re-distributed to the remaining valid users. This is a prohibitively costly task, especially in the case of encryption of cloud data due to the large amount of data stored in the particular cloud account.

[1] More precisely, in the context of PRE, first-level ciphertext and second-level ciphertext are distinguished, and the proxy can transform the delegator's second-level ciphertexts into the first-level ciphertexts of the delegatee. Hereafter in this paper, by saying that the proxy transforms ciphertexts, we are referring to the transforming of second-level ciphertexts of the delegator.

In this paper, we demonstrate how these two challenges can be addressed using CPRE that also allows us to implement cloud server-enabled user revocation. In other words, the cloud server plays the role of the proxy, and thus is given a set of re-encryption keys, each being from the data owner (who is the delegator) to an authorized user (who is the delegatee). When an authorized user requests some data from the cloud storage account, the cloud server transforms the data using the corresponding re-encryption key. User revocation is attained by deleting the revoked user's re-encryption key from the cloud server, requiring neither update of the encryption key and re-distribution of the key, nor re-generation of the cloud data.

While the above idea is promising, the majority of existing CPRE schemes (e.g. [2, 3, 7]) can only cope with simple, keyword-based conditions, i.e., both w and w' are a keyword or an "AND" concatenation of keywords. This limits the usefulness of such schemes, and is not an effective solution for enforcing fine-grained cryptographic access control required for cloud data. We are, thus, motivated to propose a new CPRE scheme that is capable of handling more fine-grained conditions. In particular, our CPRE scheme can be intuitively denoted as $E(pk_A, \cdot, \mathbb{P}) \xrightarrow{rk_{A \xrightarrow{\mathbb{A}} B}} E(pk_B, \cdot)$, where $\mathbb{P}$ is an access policy and $\mathbb{A}$ is a set of attributes. Transformation of the ciphertext can be accomplished if, and only if, $\mathbb{P}$ is satisfied by $\mathbb{A}$. Access policies $\mathbb{P}$ in our scheme are expressive and can be any monotonic access structures, comparable to those considered in attribute-based encryption (ABE) [8, 9]. Following the naming convention of ABE [8, 9], our proposal is a ciphertext-policy attribute based CPRE scheme.

Contributions. This is an extended version of [10]. We remark that the CPRE scheme presented in this paper is different from [10], as the scheme in [10] is unable to meet the security definition described in Section 3 (see the Remark in Section 4.1). However, the previous scheme is still secure for secure cloud based data sharing. We then prove the proposed scheme secure, and present our experimental evaluations that demonstrate the effectiveness of the proposed scheme.

Organization. The rest of the paper is organized as follows. In Section 2, we review related work on PRE/CPRE. Formulation of the ciphertext-policy attribute based CPRE is presented in Section 3 and a concrete scheme is given in Section 4, together with its security analysis. In Section 5 we deploy our scheme to provide fine-grained encryption of cloud data, based on the cloud-enabled user revocation approach. Section 6 presents the experimental results, and Section 7 concludes this paper.

2. Related Work

In this section, we review related work on PRE/CPRE, and defer the background information on encryption and sharing of cloud data to Section 5.2.

Blaze, Bleumer and Strauss [1] formalized the concept of PRE, and proposed the first bidirectional PRE scheme (in which the delegation from Alice to Bob also allows for re-encryption from Bob to Alice). The first unidirectional PRE scheme, due to Ateniese *et al.* [11], is based on bilinear pairings.

The schemes by Blaze, Bleumer and Strauss [1] and Ateniese *et al.* [11] are only secure against chosen-plaintext attacks (CPA). Canetti and Hohenberger [12] presented the first chosen-ciphertext attack (CCA)-secure PRE scheme based on bilinear pairings, which is a bidirectional scheme. A year later, Libert and Vergnaud [13] presented an unidirectional PRE scheme with a weaker form of CCA security, namely, security under replayable chosen-ciphertext attacks (RCCA) [14].

Earlier PRE schemes rely on costly bilinear pairings. This motivates researchers to construct PRE schemes without using bilinear pairings. Deng *et al.* [15] was, probably, the first to publish a bidirectional PRE scheme not based on bilinear pairings, and is CCA-secure bidirectional scheme. The first two unidirectional PRE schemes without bilinear pairings were due to Shao and Cao [16], and Weng *et al.* [17].

Attribute based PRE [18, 19, 20, 21, 22] implements attribute-set to attribute-set transformation (rather than individual to individual transformation), and can be viewed as an extension to the conventional PRE, with each of the delegator and the delegatee being represented by a set of attributes. Dynamic credential and ciphertext delegation for ABE [23] also uses the attribute-set to attribute-set transformation.

Conditional proxy re-encryption (CPRE) aims at restricting the transformation capability of the proxy, such that a transformation by the proxy can succeed only if the prescribed conditions are met. Earlier CPRE schemes [2, 3, 7] can only accommodate simple, keyword-based conditions, where both values embedded with the delegator's ciphertext and with the re-encryption keys are either a keyword or a "AND" concatenation of several keywords. As a result, these CPRE schemes are not capable in enforcing of finer-grained delegations of decryption rights.

In 2010, Zhao *et al.* [24] proposed an attribute-based CPRE scheme, which is the first to support fine-grained conditions more than keywords. Their scheme can be described as $E(pk_A, \cdot, \mathbb{A}) \xrightarrow{rk_{A \xrightarrow{\mathbb{P}} B}} E(pk_B, \cdot)$, where $\mathbb{A}$ is a set of attributes and $\mathbb{P}$ is an access policy. Following the naming convention of key-policy ABE [9] and ciphertext-policy ABE [8], their scheme is key-policy attribute based CPRE (as opposed to our proposed ciphertext-

policy attribute based CPRE).

However, existing key-policy attribute based schemes (e.g. [24]) suffer from two key weaknesses. (1) These schemes require one to pre-define the total number N of attributes admitted by the system (this is not practical in a cloud deployment), and the size of the system's public key is linear to N. In contrast, our scheme does not have this limitation, and any subsequent update of N does not affect the system setting. (2) Inheriting the advantages of ciphertext-policy ABE (over key-policy ABE), ciphertext-policy attribute based CPRE is more natural and flexible than its key-policy counterpart when used to encrypt cloud data, as the former allows the data owner to directly specify the access control policy associated with a ciphertext, under which the ciphertext can be decrypted.

3. Model of Ciphertext-Policy Attribute Based CPRE

In this section, we formalize the ciphertext-policy attribute based conditional proxy re-encryption model and its security notions. We achieve the same expressiveness of access policies as in ABE schemes [8, 9] and, thus, we adopt the same definitions for "policy" and "attribute".

3.1. Preliminaries

Access Tree. In ciphertext-policy attribute based CPRE, a re-encryption key is generated with a set of *descriptive attributes* $\mathbb{A}$. An encryptor wishing to encrypt a message specifies an *access policy* $\mathbb{P}$ and generates the ciphertext under $\mathbb{P}$. For a re-encryption key to be able to transform the ciphertext, its associated attributes $\mathbb{A}$ must satisfy the access policy $\mathbb{P}$. An access policy is expressed by an *access tree*, where each leaf node represents an attribute and we use $\mathsf{att}(\ell)$ to denote the attribute associated with leaf node ℓ. Each non-leaf node of the tree represents a threshold gate, described by its children and a threshold value. Let num_n be the number of children of a non-leaf node n, and t_n be its threshold value, where $1 \leq t_n \leq num_n$. When $t_n = 1$, the threshold gate is an OR gate and when $t_n = num_n$, it is an AND gate. The parent of a node n in the tree is denoted by $\mathsf{parent}(n)$. The tree also defines an ordering among the children of a node, i.e., the child nodes of a node n are numbered from 1 to num_n. The function $\mathsf{index}(n)$ calculates such a unique number associated with a node n. Access tree can express any access policy in the form of a monotonic formulae.

Satisfying an Access Tree. Let $\mathcal{T}$ be an access tree with root rt. $\mathcal{T}_n$ denotes the subtree of $\mathcal{T}$ rooted at node n. Hence $\mathcal{T} = \mathcal{T}_{rt}$. When a set $\mathbb{A}$ of attributes satisfy the access tree $\mathcal{T}_n$, it is denoted as $\mathcal{T}_n(\mathbb{A}) = 1$. $\mathcal{T}_n(\mathbb{A})$ is computed in a recursive way as

follows: if n is a non-leaf node, compute $\mathcal{T}_{n'}(\mathbb{A})$ for all child nodes n'; $\mathcal{T}_n(\mathbb{A})$ returns 1 if and only if at least t_n children return 1; if n is a leaf node, then $\mathcal{T}_n(\mathbb{A})$ returns 1 if and only if $\mathsf{att}(n) \in \mathbb{A}$.

3.2. Model

A ciphertext-policy attribute based CPRE scheme consists of the following algorithms.

Setup$(1^\kappa) \rightarrow params$**:** On input a security parameter 1^κ, the setup algorithm outputs public parameter $params$. For conciseness, below we assume that $params$ is implicitly included in the input of the rest algorithms.

KeyGen$(u) \rightarrow (pk_u, sk_u)$**:** On input a user identity U, the key generation algorithm outputs a public/private key pair pk_u/sk_u for user u.

ReKeyGen$(sk_A, \mathbb{A}, pk_B) \rightarrow rk_{A \xrightarrow{\mathbb{A}} B}$**:** On input a private key sk_A of the delegator, a set of attributes $\mathbb{A}$ and a public key pk_B of the delegatee, the re-encryption key generation algorithm outputs a re-encryption key $rk_{A \xrightarrow{\mathbb{A}} B}$ from A to B issued upon $\mathbb{A}$.

Enc$_1(pk, m) \rightarrow c$**:** On input a public key pk and a plaintext $m \in \mathcal{M}$ ($\mathcal{M}$ is an appropriate message space), the first-level encryption algorithm outputs a first-level ciphertext c.

Enc$_2(pk, m, \mathbb{P}) \rightarrow \tilde{c}$**:** On input a public key pk, a message $m \in \mathcal{M}$ and an access policy $\mathbb{P}$, the second-level encryption algorithm outputs a second-level ciphertext $\tilde{c}$. Note that in the setting of PRE, it distinguishes between first-level encryption and second-level encryption, and only second-level ciphertexts could be transformed (into first-level ciphertexts).

ReEnc$(rk_{A \xrightarrow{\mathbb{A}} B}, \tilde{c}_A) \rightarrow c_B$**:** On input a re-encryption key $rk_{A \xrightarrow{\mathbb{A}} B}$ from A to B associated with $\mathbb{A}$, and a second-level ciphertext $\tilde{c}_A$ under public key pk_A and an access policy $\mathbb{P}$, the re-encryption algorithm outputs a first-level ciphertext c_B under public key pk_B if $\mathbb{A}$ satisfies $\mathbb{P}$; otherwise, it outputs $\perp$.

Dec$_1(sk, c) \rightarrow m$**:** On input a private key sk and a first-level ciphertext c under pk, the first-level decryption algorithm outputs a plaintext m if pk/sk is a valid key pair.

Dec$_2(sk, \tilde{c}) \rightarrow m$**:** On input a private key sk and a second-level ciphertext $\tilde{c}$ under pk and a certain access policy, the second-level decryption algorithm outputs a plaintext m if pk/sk is a valid key pair.

Correctness. The correctness of a cipher-text attribute based CPRE scheme asserts that, for any $m \in \mathcal{M}$, any $\mathbb{P}, \mathbb{A}$, any key pairs pk_A/sk_A and pk_B/sk_B, the following holds if $\mathbb{A}$ satisfies $\mathbb{P}$

$$\begin{aligned}
&\mathrm{Dec}_1(sk_A, \mathrm{Enc}_1(pk_A, m))) = m \\
&\mathrm{Dec}_2(sk_A, \mathrm{Enc}_2(pk_A, m, \mathbb{P}))) = m \\
&\mathrm{Dec}_1(sk_B, \mathrm{ReEnc}(\mathrm{ReKenGen}(sk_A, \mathbb{A}, pk_B), \mathrm{Enc}_2(pk_A, m, \mathbb{P}))) \\
&= m
\end{aligned}$$

3.3. Security Notion

Security of ciphertext-policy attribute based CPRE is defined through the following security game between a challenger and an adversary $\mathcal{A}$.

Setup. The challenger runs the Setup algorithm to generate public parameter *params*, and then executes KeyGen with a random user identity O to get a key pair pk_O/sk_O. Finally, the challenger passes *params* and pk_O to the adversary $\mathcal{A}$.

Phase 1. The adversary $\mathcal{A}$ makes a number of re-encryption key generation queries on sets of attributes $\mathbb{A}_1, \mathbb{A}_2, \cdots, \mathbb{A}_{q_1}$ to the challenger. For each query (which is on $\mathbb{A}_j$), the challenger responds as follows. First, execute the KenGen algorithm with a random user identity to obtain a public/private key pair pk_j/sk_j. Then execute $\mathsf{ReKeyGen}(sk_O, \mathbb{A}_j, pk_j) = rk_{O \xrightarrow{\mathbb{A}_j} j}$. Finally, return pk_j/sk_j and $rk_{O \xrightarrow{\mathbb{A}_j} j}$ to $\mathcal{A}$.

Challenge. The adversary $\mathcal{A}$ submits two equal length messages m_0 and m_1, together with a challenge access policy $\mathbb{P}^*$ such that none of $\mathbb{A}_1, \mathbb{A}_2, \cdots, \mathbb{A}_{q_1}$ from Phase 1 satisfies $\mathbb{P}^*$. The challenger flips a random coin b, executes $\mathsf{Enc}_2(pk_O, m_b, \mathbb{P}^*) = \tilde{c}^*$, and returns the ciphertext $\tilde{c}^*$ to $\mathcal{A}$.

Phase 2. $\mathcal{A}$ again makes a number of re-encryption key generation queries on sets of attributes $\mathbb{A}_{q_1+1}, \mathbb{A}_{q_1+2}$,
$\cdots, \mathbb{A}_{q_2}$ to the challenger. For each query on $\mathbb{A}_j$, the challenger responds as follows. First, execute the KenGen algorithm with a random user identity to obtain a public/private key pair pk_j/sk_j. Then execute $\mathsf{ReKeyGen}(sk_O, \mathbb{A}_j, pk_j) = rk_{O \xrightarrow{\mathbb{A}_j} j}$. Finally, if $\mathbb{A}_j$ satisfies $\mathbb{P}^*$, then return pk_j and $rk_{O \xrightarrow{\mathbb{A}_j} j}$ to the adversary; otherwise, return pk_j/sk_j and $rk_{O \xrightarrow{\mathbb{A}_j} j}$ to $\mathcal{A}$.

Guess. The adversary $\mathcal{A}$ outputs a bit b', which is a guess on b.

Definition 1. [CPA Security] A ciphertext-policy attribute based CPRE scheme is CPA (chosen plaintext attack) secure if for any PPT adversary $\mathcal{A}$, it holds that $\Pr[b' = b] = 1/2 + \epsilon(\kappa)$, where ϵ is a negligible function.

It appears to be a common practice to separately define first-level encryption security and second-level encryption security in existing PRE/CPRE literature, e.g., [11, 3, 7]. Our formalization simultaneously captures both notations. More specifically, the challenge ciphertext $\tilde{c}^*$ is a second-level ciphertext; thus, second-level encryption security is trivially captured. In addtion, the adversary is allowed to obtain re-encryption keys whose associated attributes satisfy the challenge access policy $\mathbb{P}^*$ (in which case, $\mathcal{A}$ does not get the private keys of delegatees). Therefore, the adversary can transform the challenge ciphertext into first-level ciphertexts using these re-encryption keys, and this captures first-level encryption security.

We remark that the CPA security notion is sufficient in meeting the needs for encryption of cloud data – the intended application of CPRE in this paper. Therefore, we design our scheme to achieve only CPA security. It is also relatively straightforward to extend the scheme to achieve CCA security using generic techniques for constructing CCA secure schemes (e.g. [25, 26, 27]).

4. Our Scheme

In this section, we present our construction of the ciphertext-policy attribute based CPRE scheme, as well as its security analysis.

4.1. The Construction

Our construction is inspired by the ciphertext-policy ABE scheme [8], but there remains a big gap between the ABE scheme and our ciphertext-policy attribute based CPRE scheme (consider the gap between public key encryption and PRE).

Let $s \in_R S$ denote an element s randomly drawn from a set S. The details of our scheme are as follows.

Setup(1^κ)**:** Determine a bilinear map $e : G_0 \times G_0 \to G_T$, where G_0 and G_T are cyclic groups of κ-bit prime order p; select g which is a generators of G_0, a cryptographic hash function $H : \{0,1\}^* \to G_0$. Then set $params = (g, \mathcal{G} = e(g,g), H(\cdot))$.

KeyGen(u)**:** Pick $x_u, \alpha_u, \beta_u \in_R Z_p$, and set $(pk_u = \langle y_u = g^{x_u}, h_u = g^{\beta_u}, w_u = \mathcal{G}^{\alpha_u} \rangle, sk_u = \langle x_u, \alpha_u, \beta_u \rangle)$.

ReKeyGen$(sk_A, \mathbb{A}, pk_B)$**:** Let $sk_A = \langle x_A, \alpha_A, \beta_A \rangle$ and $pk_B = \langle y_B = g^{x_B}, h_B = g^{\beta_B}, w_B = \mathcal{G}^{\alpha_B} \rangle$. Pick $r, r_i \in_R Z_p, \forall i \in \mathbb{A}$, and set the re-encryption key as

$$rk_{A \xrightarrow{\mathbb{A}} B} = (k = (y_B^{\alpha_A} g^r)^{\frac{1}{\beta_A}}, \forall i \in \mathbb{A} : \{k_{i1} = g^r H(i)^{r_i}, k_{i2} = g^{r_i}\})$$

Enc$_1(pk, m)$**:** Let $pk = \langle y = g^x, h = g^\beta, w = \mathcal{G}^\alpha \rangle$. Pick $s \in_R Z_p$ and $w' \in_R G_0$; then compute

$$c = (m \cdot e(w', g)^s, e(w', y)^s = e(w', g)^{s.x})$$

Enc$_2(pk, m, \mathbb{P})$**:** Let $pk = \langle y = g^x, h = g^\beta, w = \mathcal{G}^\alpha \rangle$ and $\mathbb{P}$ be an access tree $\mathcal{T}$. The second-level encryption algorithm proceeds as follows. It first selects a polynomial q_n for each node n (including the leaf nodes) in $\mathcal{T}$. These polynomials are chosen in a top-down manner, starting from the root node rt: for each node n, set the degree d_n of the polynomial q_n to be $d_n = t_n - 1$, where t_n is the threshold value of node n. Starting with the root node rt the algorithm selects a random $s \in_R Z_p$ and sets $q_{rt}(0) = s$. It then selects d_{rt} other random points to define q_{rt}. For any other node n, it sets $q_n(0) = q_{\mathsf{parent}(n)}(\mathsf{index}(n))$ and chooses d_n other points to define q_n.

Let L be the set of leaf nodes in $\mathcal{T}$. Sets the second level ciphertext $\tilde{c}$ as

$$\tilde{c} = (\mathcal{T}, C = m \cdot w^s, C' = h^s, \forall l \in L : \{C_{l1} = g^{q_l(0)}, C_{l2} = H(\mathsf{att}(l))^{q_l(0)}\})$$

ReEnc$(rk_{A \xrightarrow{\mathbb{A}} B}, \tilde{c}_A)$**:** Let $\tilde{c}_A = (\mathcal{T}, C, C', \forall l \in L : \{C_{l1}, C_{l2}\})$ and $rk_{A \xrightarrow{\mathbb{A}} B} = (k, \forall i \in \mathbb{A} : \{k_{i1}, k_{i2}\})$. The re-encryption algorithm is a recursive procedure. We first define an algorithm $\mathsf{ReEncNd}_n(\tilde{c}_A, rk_{A \xrightarrow{\mathbb{A}} B})$ on a node n of $\mathcal{T}$. If node n is a leaf node, then we let $z = \mathsf{att}(n)$ and define as follows: $z \notin \mathbb{A}$, then $\mathsf{ReEncNd}_n(\tilde{c}_A, rk_{A \xrightarrow{\mathbb{A}} B}) = \perp$; otherwise $\mathrm{ReEncNd}_n(\tilde{c}_A, rk_{A \xrightarrow{\mathbb{A}} B}) = F_n$, where

$$\begin{aligned} F_n &= \frac{e(k_{z1}, C_{n1})}{e(k_{z2}, C_{n2})} \\ &= \frac{e(g^r H(z)^{r_z}, g^{q_n(0)})}{e(g^{r_z}, H(z)^{q_n(0)})} \\ &= e(g, g)^{r.q_n(0)} \end{aligned} \tag{1}$$

Let us now consider the recursive case when n is a non-leaf node. The algorithm $\mathsf{ReEncNd}_n(\tilde{c}_A, rk_{A \xrightarrow{\mathbb{A}} B})$ proceeds as follows. For each child node v of n, it calls $\mathsf{ReEncNd}_v(\tilde{c}_A, rk_{A \xrightarrow{\mathbb{A}} B})$ and stores the output as F_v. Let S_n be an arbitrary t_n-sized set of child nodes v such that $F_v \neq \perp$. If such a set does not exist, then the node is not satisfied and $\mathrm{ReEncNd}_n(\tilde{c}_A, rk_{A \xrightarrow{\mathbb{A}} B}) = F_n = \perp$. Otherwise, let the Lagrange

coefficient $\triangle_{i,S}$ for $i \in Z_p$ and a set S of elements in Z_p be $\triangle_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j}$, and we compute

$$\begin{aligned} F_n &= \prod_{v \in S_n} F_v^{\triangle_{i,S'_n}(0)}, \text{where } {}^{i=\mathsf{index}(v),}_{S'_n=\{\mathsf{index}(v): v \in S_n\}} \\ &= \prod_{v \in S_n} (e(g,g)^{r \cdot q_v(0)})^{\triangle_{i,S'_n}(0)} \\ &= \prod_{v \in S_n} (e(g,g)^{r \cdot q_{\mathrm{parent}(v)}(\mathrm{index}(v))})^{\triangle_{i,S'_n}(0)} \\ &= \prod_{v \in S_n} (e(g,g)^{r \cdot q_n(i)})^{\triangle_{i,S'_n}(0)} \\ &= e(g,g)^{r \cdot q_n(0)} \end{aligned} \quad (2)$$

In this way, $\mathsf{ReEncNd}_{rt}(\tilde{c}_A, rk_{A \xrightarrow{\mathbb{A}} B})$ for the root node rt can be computed. If $\mathcal{T}_{rt}(\mathbb{A}) = 1$, then we get $\mathsf{ReEncNd}_{rt}(\tilde{c}_A, rk_{A \xrightarrow{\mathbb{A}} B}) = e(g,g)^{r \cdot q_{rt}(0)} = e(g,g)^{r \cdot s} = F_{rt}$. Next, the re-encryption algorithm continues to compute $e(k, C')/F_{rt} = e((y_B^{\alpha_A} g^r)^{\frac{1}{\beta_A}}, h_A^s)/e(g,g)^{r \cdot s} = e(y_B, g)^{s \cdot \alpha_A} = w_A^{s \cdot x_B}$, where $w_A = \mathcal{G}^{\alpha_A}$. Finally, sets $c_B = (c_1 = C = m \cdot w_A^s, c_2 = w_A^{s \cdot x_B})$[2].

$\mathbf{Dec}_1(sk, c)$: Let $sk = \langle x, \alpha, \beta \rangle, c = (c_1, c_2)$, then decryption of the first level ciphertext is $m = c_1 / c_2^{x^{-1}}$.

$\mathbf{Dec}_2(sk, \tilde{c})$: Let $sk = \langle x, \alpha, \beta \rangle, \tilde{c} = (\mathcal{T}, C, C', \cdots)$, decryption of the second level ciphertext is $m = C / e(g, C')^{\alpha/\beta^{-1}}$.

Remark. We remark that this scheme differs from our previous scheme [10], as the latter does not satisfy the CPA definition (see Definition 1). In the previous scheme [10], F_n in the ReEnc algorithm (which corresponds to the above Eqn. (1)) is of the form $F_n = e(g,g)^{x_B \cdot q_n(0)}$, where x_B is the private key of the delegatee. As such the delegatee is able to cancel out the x_B element from F_n by computing $F'_n = F_n^{x_B^{-1}}$. As a result, delegatees can pool their F'_n's by Lagrange interpolation as in Eqn. (2). However, it seems still safe to apply the scheme in [10] to the encryption of cloud data, as the cloud server in the Data Access algorithm simply returns $\perp$ if a requesting user's re-encryption key does not satisfy the requested ciphertext's access policy, and users are not able to pool their transformed ciphertexts.

4.2. Security Analysis

The security of our proposed scheme is defined in Theorem 4.1.

[2] If we see w_A as an ordinary element in G_T (i.e., not a part of the delegator's public key), the format of the transformed ciphertext conforms exactly to that of the first-level ciphertext.

Theorem 4.1. *The above scheme is CPA secure (in the sense of Definition 1) in the generic group model [28].*

Proof The proof considers two separate two cases: the first case is that $\mathcal{A}$ does not make a re-encryption key generation query with an attribute set $\mathbb{A}_i$ satisfying the challenge access policy $\mathbb{P}^*$, and the second is that $\mathcal{A}$ makes at least one re-encryption key generation query whose attribute set $\mathbb{A}_i$ satisfies $\mathbb{P}^*$. Note that for each of the first type re-encryption key generation queries, $\mathcal{A}$ will obtain the delegatee's private key, together with the re-encryption key; while for the second type, $\mathcal{A}$ will not obtain the delegatees' private keys.

For the first case, we prove that $\mathcal{A}$ for the above scheme is directly reduced to an adversary $\mathcal{B}$ for the ciphertext-policy ABE scheme [8], which is proven secure in the generic group model. To this end, $\mathcal{A}$ needs to simulate a game to $\mathcal{B}$ with respect to the ciphertext-policy ABE scheme. In particular, $\mathcal{A}$ has to answer $\mathcal{B}$'s decryption key generation queries (for each query, $\mathcal{B}$ submits an attribute set). We show that within the context of the CPA game between $\mathcal{A}$ and its challenger, $\mathcal{A}$ can answer $\mathcal{B}$'s decryption key generation queries. Specifically, receiving an attribute set $\mathbb{A}$ from $\mathcal{B}$, $\mathcal{A}$ makes a re-encryption key query on $\mathbb{A}$ to its challenger, and subsequently obtains the re-encryption key $(k, \{k_{i1}, k_{i2}\}_{i\in\mathbb{A}})$ as well as the delegatee's public/private key pk/sk (by assumption, $\mathcal{A}$ will obtain the delegatee's private key $sk = (x, \alpha, \beta)$). Then $\mathcal{A}$ computes and returns $(k^{x^{-1}}, \{k_{i1}^{x^{-1}}, k_{i2}^{x^{-1}}\}_{i\in\mathbb{A}})$ to $\mathcal{B}$ as the reply. Note that $(k^{x^{-1}}, \{k_{i1}^{x^{-1}}, k_{i2}^{x^{-1}}\}_{i\in\mathbb{A}})$ is, indeed, a valid decryption key for the ciphertext-policy ABE scheme in [8], with r/x and r_i/x acting as random numbers. It is trivial to observe that the simulation by $\mathcal{A}$ is perfect.

In the second case, for simplicity we assume that the attributes $\mathbb{A}_i$ in every re-encryption key generation query of $\mathcal{A}$ satisfy the challenge access policy $\mathbb{P}^*$. We prove that under the generic group model, no efficient adversary $\mathcal{A}$ can guess $b' = b$ in the security game defined in Definition 1 noticeably better than a random guess. Note that a random guess b' by $\mathcal{A}$ equals b with probability 1/2, and the advantage of $\mathcal{A}$ is denoted as ϵ.

In the generic group model, elements of groups G_0, G_T are encoded as unique random strings and, thus, the adversary can directly test no property other than equality. The opaque encoding of the elements in G_0 is defined as the function $\xi_0 : Z_p \to \{0,1\}^*$, which maps all $a \in Z_p$ to the string representation $\xi_0(a)$ of $g^a \in G$. Likewise, $\xi_T : Z_p \to \{0,1\}^*$ maps $a \in Z_p$ to the string representation $\xi_T(a)$ of $e(g,g)^a \in G_T$. The adversary $\mathcal{A}$ communicates with oracles to compute group action on G_0, G_T and bilinear map $e : G_0 \times G_0 \to G_T$, through the ξ_0-representation and ξ_T-representation only.

For simplicity, we modify the game slightly: in the challenge phase of the security game, $\mathcal{A}$ is given a challenge ciphertext whose C component is either $m_0 \cdot e(g,g)^{\alpha\cdot s}$ or

$m_1 \cdot e(g,g)^{\alpha \cdot s}$. We modify C to be either $e(g,g)^{\alpha \cdot s}$ or $e(g,g)^{\vartheta}$, for a random ϑ in Z_p. Indeed, any adversary $\mathcal{A}$ that has advantage ϵ in the original security game can be transformed into an adversary having an advantage of $\epsilon/2$ in the modified game (consider two hybrids: one in which the adversary is to distinguish between $m_0 \cdot e(g,g)^{\alpha \cdot s}, e(g,g)^{\vartheta}$), and the other in which the adversary is to distinguish between $m_1 \cdot e(g,g)^{\alpha \cdot s}, e(g,g)^{\vartheta}$).

Hereafter, we consider $\mathcal{A}$ in a modified game. In the Setup phase, the challenger sends the public parameters $\xi_0(1), \xi_T(1)$ to $\mathcal{A}$. To simulate the hash function H, the challenger maintains a table, which is initially empty. Whenever $\mathcal{A}$ asks a query i on H, if i has never been asked before, the challenger selects a random value $t_i \in_R G_T$, and adds an entry $(i, t_i, \xi_0(t_i))$ to the table and returns $\xi_0(t_i)$ to $\mathcal{A}$; otherwise, returns the already defined $\xi_0(t_i)$ to $\mathcal{A}$.

In Phase 1, for a jth re-encryption key generation query $\mathbb{A}$, the challenger first picks x_A, α_A, β_A and sets $pk_A = \langle y_A = \xi_0(x_A), h_A = \xi_0(\beta_A), w_A = \xi_0(\alpha_A) \rangle, sk_A = \langle x_A, \alpha_A, \beta_A \rangle$, and picks x_B, α_B, β_B and sets $pk_B = \langle y_B = \xi_0(x_B), h_B = \xi_0(\beta_B), w_B = \xi_0(\alpha_B) \rangle, sk_B = \langle x_B, \alpha_B, \beta_B \rangle$. Then selects $r^{[j]}, r_i^{[j]} \in_R Z_p, i \in \mathbb{A}$, and returns $pk_A, pk_B, rk = \langle k = \xi_0(\frac{x_B \alpha_A + r^{[j]}}{\beta_B}), \forall i \in \mathbb{A} : \{k_{i1} = \xi_0(r^{[j]} + t_i r_i^{[j]}), k_{i2} = \xi_0(r_i^{[j]})\} \rangle$ to $\mathcal{A}$.

In the Challenge phase, $\mathcal{A}$ submits two equal-length challenge messages m_0, m_1 and a challenge access policy $\mathbb{P}^*$. The challenger responds as follows. Selects $s \in_R Z_p$, and computes shares ς_i of s for all attributes i contained in $\mathbb{P}^*$ according to the secret sharing as described in the Enc_2 algorithm. Note that ς_i's are random values subject to the underlying secret sharing scheme induced by $\mathbb{P}^*$. Finally, the challenger chooses $\vartheta \in_R Z_p$, and returns to $\mathcal{A}$ the challenged ciphertext as $C = \xi_T(\vartheta)$ and $C' = \xi_0(\beta \cdot s), C_{i1} = \xi_0(\varsigma_i), C_{i2} = \xi_0(t_i \varsigma_i)$ for each relevant attribute i in $\mathbb{P}^*$.

In Phase 2, the challenger continues to respond to re-encryption key generation queries from $\mathcal{A}$, just as in Phase 1.

Analysis of the Simulated Game. Let q bound the total number of group elements $\mathcal{A}$ receives during the game from the queries it makes to the oracles for G_0, G_T, the bilinear map, and the hash function (including the hash function queries implicitly invoked by re-encryption key generation and challenge ciphertext generation). We will show that with probability $1 - \mathcal{O}(q^2/p)$, the adversary's view in this simulated game is identically distributed to its view if it has been given $C = \xi_T(\alpha \cdot s)$ in the game. Note that in the current game, the adversary's advantage is 0, as $\xi_T(\vartheta)$ is independent of the encryption of the challenge messages. We, thus, conclude that the advantage of the adversary, when given $C = \xi_T(\alpha \cdot s)$, is at most $\mathcal{O}(q^2/p)$, which proves the theorem if q^2/p is negligible.

We assume that $\mathcal{A}$ communicates with group oracles, only with values it has already

received from the oracles. Note that each query $\mathcal{A}$ makes is of the form of a rational function $\pi = \chi/\gamma$ in the variables of $x_A, x_B, \alpha_A, \alpha_B, \beta_A, \beta_B, t_i, r^{[j]}, r_i^{[j]}$, and ς_i. We now condition on the event that no "unexpected collisions" occur in either G_0 and G_T. An unexpected collision would be when two queries of two distinct rational functions $\chi/\gamma \neq \chi'/\gamma'$, due to the random choices of the values of the involved variables, coincide in value. For any pair of queries corresponding to χ/γ and χ'/γ', a collision occurs only if the non-zero polynomial $\chi/\gamma - \chi'/\gamma'$ evaluates to be zero. In our case the total degree of $\chi/\gamma - \chi'/\gamma'$ is at most 5; thus the probability of a collision is $\mathcal{O}(1/p)$ [29, 30]. By a union bound, the probability of any unexpected collision happens is at most $\mathcal{O}(q^2/p)$ for q queries. As a result, we have probability $1 - \mathcal{O}(q^2/p)$ that no unexpected collisions happen.

Subject to the condition of no unexpected collisions, we need to show that $\mathcal{A}$'s view is identically distributed if the challenger has set $\vartheta = \alpha \cdot s$. To this end, note that the only way that the view of $\mathcal{A}$ can differ in the case of $\vartheta = \alpha \cdot s$ is if there are two queries π, π' into G_T such that $\pi \neq \pi'$ but $\pi \mid_{\vartheta=\alpha\cdot s} = \pi' \mid_{\vartheta=\alpha\cdot s}$. We will now show that this will not happen. Recall that ϑ only occurs as $\xi_T(\vartheta)$ which is an element of G_T; thus, the only difference that π and π' can have on ϑ is such that $\pi - \pi' = \eta\vartheta - \eta\alpha \cdot s$, for a constant η. It suffices to show that $\mathcal{A}$ can never construct a query for $\xi_T(\eta\alpha \cdot s = \pi - \pi' + \eta\vartheta)$, given that no unexpected collisions occur. This arrives at a contradiction and, therefore, proves the theorem.

This follows from the following analysis, based on the information given to $\mathcal{A}$ during the game. For ease of reference, Table 1 enumerates all rational functions in G_0 known to $\mathcal{A}$ by means of re-encryption key generation queries and the challenge ciphertext query (i, i' are possible attribute strings, and j, j' are the indices of the re-encryption key generation queries).

Re-encryption key queries	x_A	β_A	x_B	β_B
	$\frac{x_B\alpha_A + r^{[j]}}{\beta_B}$	$r^{[j]} + t_i r_i^{[j]}$	$r_i^{[j]}$	
Challenge ciphertext	$\beta_B \cdot s$	s	$t_{i'} s$	

Table 1: Rational functions in G_0

Note that since we assume that the attributes in $\mathcal{A}$'s re-encryption key generation queries satisfy the challenge access policy, we directly use s to replace ς_i's for the challenge ciphertext, because compared to ς_i's, s clearly helps $\mathcal{A}$ more in terms of gaining advantage in the game.

In addition, $\mathcal{A}$ knows rational function t_i through the hash function queries. Any query

in G_T is a linear combination of these rational functions. Observe from the table that the only rational function containing α_A is $\frac{x_B\alpha_A+r^{[j]}}{\beta_B}$. In order for $\mathcal{A}$ to produce a rational function containing $\eta\alpha_A \cdot s$ while canceling out other elements as much as possible is by pairing $\frac{x_B\alpha_A+r^{[j]}}{\beta_B}$ and $\beta_B \cdot s$. This will create a polynomial of the form $x_B\alpha_A s + r^{[j]}s$. It can be seen that the element x_B cannot be canceled out by pairing any rational functions in Table 1, as $\mathcal{A}$ does not know x_B and x_B^{-1} (note that we are working under the condition of no unexpected collisions). This completes the proof for the second case, and in turn the entire proof. □

5. An Application: Revocable Fine-grained Data Sharing Over Cloud

In this section, we present an application of our scheme: fine-grained data sharing over cloud storage services. We begin with a discussion of the application scenario.

5.1. Application Scenario

We consider the following application scenario. To enjoy the advantages of cloud storage services, a Data Owner (DO) establishes a cloud-based data sharing platform, where the DO outsources her data to the cloud storage, and authorizes a group of users to access the data, as shown in Figure 1. In particular, users are distinguished by their

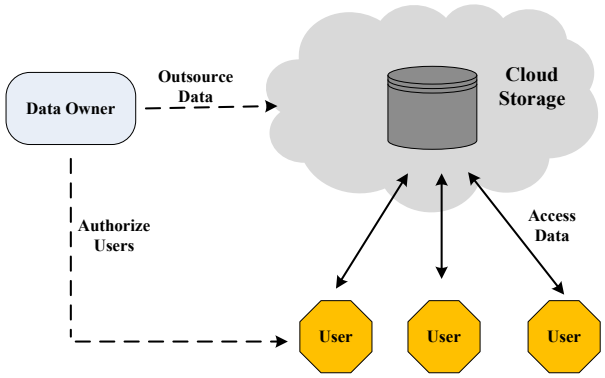

Figure 1: Application scenario of data sharing over cloud

functional roles (a functional role can be defined by a set of attributes), and are granted access right according to their respective roles. The DO is to enforce fine-grained control over the cloud data, such that data are accessed by roles on a need-to-know basis (i.e., different roles have different access rights).

However, the DO does not fully trust the cloud, and wants to ensure the privacy of the data stored in the cloud. Thus, the data uploaded to the cloud are encrypted by the DO. Each authorized user is granted appropriate decryption capability by the DO as per

the role. The system must support user revocation. Therefore, once an authorized user is revoked, the user's decryption capability will also be removed in real-time.

Example. An example of DO and authorized users are a company and its employees, respectively. The company stores the corporate data in the cloud, and allows employees to access data on a need-to-know basis. In such a context, the company often needs to specify data access control policies based on the roles that employees assume (rather than on the identities of the employees). For example, an access control policy would be "this data can only be accessed by the executive manager from the purchasing department". As a result, the employees are distinguished by their roles, i.e., each user should be issued a *role certificate*, rather than *identity certificate*. The need for user revocation occurs, when, e.g., an employee leaves the company. In such a case, the role of the employee remains, but the rights of the individual employee are revoked. Thus, we are concerned about user revocation, rather than attribute revocation.

5.2. Encryption of Cloud Data with ABE, and User Revocation

Attribute-based encryption (ABE) (e.g. [8, 9, 31]) can potentially satisfy the needs of enforcing fine-grained cryptographic access control over cloud data in the above application scenario. A number of proposals in the literature have, for example, suggested applying ABE to encrypting cloud data. Compared to conventional public-key encryption, where a message encrypted under a public key can only be decrypted by a single corresponding private key, ABE offers much finer-grained encryption/decryption capabilities. More specifically, ABE is a one-to-many encryption primitive, whereby data are encrypted with a system-wide public key under certain access policy/attributes and each decryption key is attached to certain attributes/access policy, such that the ciphertext can be decrypted only if the attributes satisfy the access policy.

Although ABE appears promising to be used in the encryption of cloud data, user revocation remains a limiting factor. The challenge is due to the fact that ABE is essentially "group encryption"; thus, ABE does not differentiate individual users. The ABE schemes proposed by [8, 9, 31] include an "expiry time" attribute in the attribute set, such that a decryption key can work only up to its associated expiry time. However, such a mechanism cannot implement real-time revocation of decryption capabilities. The ABE scheme in [32] supports *negative* constrains in the access policy, which provides a means for attribute revocation by negating the attributes to be revoked. Such a "negation" mechanism is not scalable for user revocation in encryption of cloud data, as each revoked user must be explicitly treated as a distinct attribute and included in the ciphertext. Therefore, the scheme is not practical when the number of revoked users grows large. Other variants of

ABE, such as predicate encryption (e.g., [33]) and functional encryption (e.g., [34, 35]) are also restricted to these two revocation mechanisms.

There also exist "stateful" user revocation mechanisms, which involve updating of the decryption keys (e.g. [36, 37, 38]) that would allow re-generation of cloud data and key re-distribution. For example, [38] proposed applying KP-ABE [9] to be used in the encryption of cloud data, which also adopts the specific technique of PRE scheme [1] to update users' decryption keys for user revocation. The advantage of such an user revocation approach is that the majority of re-generation of cloud data and re-distribution of new key activities are moved to the cloud. While such an approach is a significant improvement over the trivial solution (i.e. in schemes where encryptor is entirely responsible for data re-generation and key re-distribution), it would be preferable to move the entire re-generation of cloud data and re-distribution of new key activities to the cloud. The user revocation method described in [23] is one such example, where the cloud is responsible for re-generating encrypted cloud data for user revocation.

5.3. Cloud-Enabled User Revocation

It is clear from the above discussions that it is not yet practical to deploy ABE for encryption and sharing of cloud data, until a more satisfactory user revocation solution is available. Based on our ciphertext-policy attribute based CPRE scheme, we propose a cloud-enabled user revocation approach. The basic idea of our approach is as follows. The cloud is enlisted as the proxy in CPRE, and holds the re-encryption keys of all authorized users. As such, the cloud is enabled to transform encrypted data records into those under a user's public key. If the user is authorized and, therefore, will have a re-encryption key held by the cloud. To revoke a user, we would simply erase the user's re-encryption key from the cloud and, thus, the cloud will not be able to transform encrypted data for the user. Note that re-encryption key erasure could either be undertaken by the data owner (DO) through an administration interface or by the cloud under the DO's instruction. Using this approach, user revocation incurs little cost, and requires no key update, no re-generation of cloud data, and no re-distribution of keys.

A comparative summary is outlined in Table 2.

5.4. Application Details

We now explain how our proposed CPRE scheme can be deployed for encryption of cloud data.

System Setup. The DO runs CPRE.Setup to establish and publish the public parameter of the CPRE scheme.

Table 2: A comparison of user revocation approaches

	Immediate Revocation	No Key update & data re-gen.	Scalability
Expiry time	×	✓	✓
Negation	✓	✓	×
[36, 37, 38]	✓	×	✓
Our approach	✓	✓	✓

✓: Can be achieved; ×: Unable to be achieved

The DO runs $\mathsf{CPRE.KeyGen}$ to generate his/her own key pair pk_O/sk_O. Similarly, users u generate their own key pairs pk_u/sk_u. All public keys are part of the public system parameters.

User Authorization. To authorize a user u, the DO decides a set of attributes $\mathbb{A}$ according to u's role; executes $rk_{O \xrightarrow{\mathbb{A}} u} = \mathsf{CPRE.ReKeyGen}(sk_O, \mathbb{A}, pk_u)$, and sends $rk_{O \xrightarrow{\mathbb{A}} u}$ to the cloud in a secure manner. Note that $rk_{O \xrightarrow{\mathbb{A}} u}$ is denoted as the role certificate for u. The cloud maintains a Re-encryption Key Table (RKT) with each entry being a user identity along with his re-encryption key, as shown below. As a result, the cloud adds a new entry $(u, rk_{O \xrightarrow{\mathbb{A}} u})$ to this re-encryption key table.

User Identity	Re-encryption Key
Alice	$rk_{O \xrightarrow{\mathbb{A}_A} A}$
Bob	$rk_{O \xrightarrow{\mathbb{A}_B} B}$
John	$rk_{O \xrightarrow{\mathbb{A}_J} J}$
Mary	$rk_{O \xrightarrow{\mathbb{A}_M} M}$
...	...

The cloud manages a RKT, which contains the re-encryption keys of all authorized users.

Data Outsourcing. To outsource a data record m to the cloud, the DO determines the access policy $\mathbb{P}$ of the record, and then computes $\tilde{c} = \mathsf{CPRE.Enc}_2(pk_O, m, \mathbb{P})$. Finally, the DO uploads $\tilde{c}$ to the cloud.

Data Access. Suppose a user u wishes to retrieve an encrypted data record $\tilde{c}$ from the cloud, the cloud first obtains u's re-encryption key $rk_{O \xrightarrow{\mathbb{A}_u} u}$ from the RKT (Re-encryption Key Table), and then computes $c_u = \mathsf{CPRE.ReEnc}(rk_{O \xrightarrow{\mathbb{A}_u} u}, \tilde{c})$ before sending it to user u. At the user side, u obtains m by computing $m = \mathsf{CPRE.Dec}_1(sk_u, c_u)$.

User Revocation. To revoke an authorized user u, the re-encryption key of u is simply erased from the cloud's re-encryption key table. Depending on the specific application, either the cloud updates the re-encryption key table (as instructed by the DO), or a management interface is provided to the DO so that the DO can manage the deletion.

In short, the DO encrypts cloud data using the Enc_2 algorithm, and decryption by the user is performed using the Dec_1 algorithm. Enc_1 and Dec_2 are not used in the application.

In this application, we assume that the cloud is semi-trusted (i.e. the cloud is not to be trusted in ensuring the secrecy of the stored data, but it can be trusted to manage the cloud data, and process user access requests and other administrative activities). This is a similar assumption on the trust invested in the proxy in the CPRE setting. We also assume that when a user requests data from the cloud, the user needs to include the identity in the request. This will facilitate the cloud in determining the corresponding re-encryption key to use from the Re-encryption Key Table. However, the cloud is not required to check the authenticity of the request. In other words, the cloud is not concerned about a legitimate user impersonating another. This is because each user's re-encryption key is part of an unique key pair.

The novelty of our proposal is that we utilize PRE/CPRE for user revocation, and the CPRE scheme we proposed attains fine-grained cryptographic access control, with granularity comparable to ABE.

5.5. Security Considerations

The primary security concern in the above application is to ensure the privacy of data against the (semi-trusted) cloud, which is the key motivation for the encryption of cloud data (see Section 1). In our context, the cloud plays the role of proxy in the CPRE scheme, and as we have shown in Section 4.2, the proxy in our CPRE scheme is not able to learn any useful information about the plaintext message. Hence, the application is secure in the sense of ensuring the privavcy of the data against the (semi-trusted) cloud.

5.6. Benefits

A key benefit of the application of our CPRE scheme for encryption and sharing of cloud data is user-side efficiency. Recall that our CPRE scheme is based on bilinear map and achieves fine-grained cryptographic access control, which is comparable to ABE. At the user side, however, the user only needs to receive and decrypt a first-level ciphertext in G_T. Note that G_T is an ordinary Abelian group (compared to bilinear groups). This results in high user-side efficiency in terms of both communication and computation.

Specifically, let $|G_T|$ denotes the bit length of an element in G_T. Then the size of a first-level ciphertext is $2|G_T|$, and Dec_1 involves only one exponentiation in G_T, and both are constant. In other words, user-side computation is lightweight and does not involve pairing operation (despite the fact that our scheme is built on bilinear map). This is a key difference between our proposal and existing proposals (that apply ABE for encryption

of cloud data), which rquire pairing operations to be performed by the user. The high efficiency and no pairing requirement for users make it possible to access cloud data using resource-constrained devices, such as smart mobile phones, tablets, and potentially in the near future, Internet-of-Things (IOT) devices.

The second key benefit of our proposal is the efficiency of cloud-enabled user revocation. In the case of user revocation, our proposal only requires the removal of the revoked user's re-encryption key from the cloud, without the need for key update, re-generation of cloud data, and re-distribution of keys (as required in other existing schemes).

In our proposal, cloud-enabled user revocation by CPRE is realized via the *split of decryption capability*, where decryption of a ciphertext requires the cooperation between the cloud and a delegatee. Such a feature differs from that of the *decryption key split* feature in [39]. The entity performing the decryption split (i.e. the DO in our application context) does not necessarily know the secret of a delegatee, while the entity for decryption key split in [39] needs to know the delegatee's secret. We regard this as the third key benefit of our proposal – the decryption capability split mechanism in our proposal allows the delegatee to maintain the secrecy of his/her secret, without the need to reveal it to the entity performing the decryption split. We remark that the method for delegating the workload of ABE decryption in [40] and [41] could also be applied to realize cloud-enabled user revocation, but this will result in decryption key split.

5.7. Disadvantages

We stress that our application of CPRE for implementing cloud server-enabled user revocation does not come at no cost. This approach requires the cloud platform to perform the ReEnc operation upon every data access request, which can be computationally expensive. Indeed, such online computations at the cloud side are inevitable, but the costs could be lowered using the following approach.

The actual deployment of the CPRE scheme should follow the common practice for public-key encryption, namely, the payload data is encrypted with a random key under symmetric encryption and the encryption key is encapsulated with the public-key encryption (i.e. the Enc_2 algorithm in our case). This allows us to amortize the overhead incurred by the key encapsulation. Specifically, all data sharing the same access policy are encrypted using a common symmetric encryption key. Consequently, a ReEnc operation by the cloud may allow a user to access many data records (rather than a ReEnc operation per data record). We stress that this does not necessarily require the data owner to manage the symmetric encryption keys. Instead, the data owner can obtain the encryption key corresponding to an access policy by retrieving and decrypting (i.e. Enc_2) the related

key encapsulation ciphertext from the cloud, if the access policy in question has already been used.

The overhead-amortization mechanism must support user revocation. Therefore, once a user is revoked, the data owner will use a new key for every access policy when encrypting new data records (but leave the old encrypted data records intact). This guarantees that the newly generated cloud records cannot be decrypted by the revoked user. In principle, it is not a concern that a revoked user can decrypt the data the user had been entitled to before his revocation (e.g. the user would likely have saved a copy of the data in another location).

Caveat. It may appear that the schemes in [36, 37, 38] implementing "stateful" user revocation mechanisms have an advantage over our application, in terms of the online computations by the cloud platform. In those schemes, few online computations are required by the cloud platform to respond to users' data accesses. While it seems that the high workload entrusted to the cloud due to user revocation in those schemes could be performed offline; in practice, it would be hard to arrange for the re-generation of cloud data to be offline without affecting user online data requests (consider that a user requests data while re-generation of cloud data is in progress).

More importantly, the disadvantage of online computation in our application can be understood to be a tradeoff for the unique feature of user-side efficiency and the no-requirement of pairing support at the user side.

6. Evaluation

We implemented the proposed scheme to evaluate its performance. The implementation is coded in Java, and the cryptographic algorithms are implemented using the Bouncy Castle Java Crypto Library[3]. We instantiated the bilinear map in the scheme with a 512-bit supersingular curve with embedding degree 2.

6.1. Experimental Results

In our scheme, the computation costs for Enc_2 and ReEnc are dependent on the complexity of the access control policy, and are linear to the number of leaf nodes of the access tree (more precisely, the cost for ReEnc also depends on the attributes associated with the re-encryption key).

To empirically evaluate the computational overhead of the Enc_2 and ReEnc algorithms, we run them on a PC with 2.66 GHz Intel Core2Duo and 3.25GB RAM, while trying to

[3] http://www.bouncycastle.org/java.html

avoid compounding factors such as network latency, instability of on-demand computing. We experimented with a set of access conditions whose access trees are constructed with all non-leaf nodes being AND gates, and re-encryption keys contain all attributes. Access conditions in such a form ensure that all attributes in an access tree are used in the ReEnc algorithm; thus, imposing the heaviest workload on the algorithm (i.e. the worse-case scenario). The experimental results are depicted in Figure 2, which shows, respectively, the timing of Enc_2 and ReEnc varying with the number of attributes. The timing averages over 100 runs of each experiment.

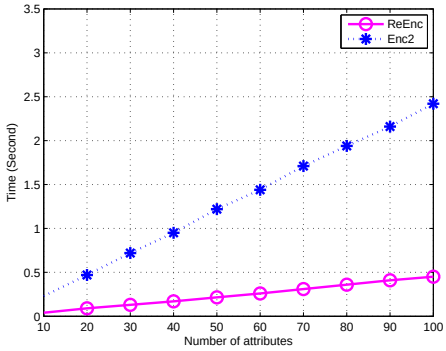

Figure 2: Experimental results on the performance of Enc_2 and ReEnc

As illustrated in Figure 2, the results corroborate the fact that both algorithms perform linear computations with the number of attributes. The computational costs of ReEnc with 60 attributes are about 1.5 seconds. This performance is realistic for real-world applications, and it should suffice to specify functional role based access control policies with 50-60 attributes.

We also implemented and tested the Dec_1 algorithm of our scheme on a smartphone, which has a 1.2 GHz CPU and 2 GB RAM. The experimental results indicate that on average, it takes about 10 milliseconds to decrypt a ciphertext. In addition, the communication overhead for the user is $2|G_T|$, about 1.2Kbits in our implementation. These results suggest that it is practical for a smartphone user to access the cloud storage.

7. Conclusion

Encryption of cloud data by the data owner is now commonly accepted as necessary to achieve data privacy against a third-party (semi-honest) cloud provider. ABE is a viable solution, as it allows fine-grained cryptographic access control, only if there is a satisfactory solution to the user revocation issue. Therefore, to address the the user revocation challenge, we proposed a ciphertext-policy attribute based CPRE scheme, which

was utilized in the implementing of the cloud-enabled user revocation approach. This application achieves access control over cloud data with granularity comparable to ABE, yet without incurring any cost for user revocation. Our proposal has applications in many real-world situations.

Another benefit of our proposal is the efficiency in terms of low computation and communication overheads required at the user side, which makes it possible for users to use resource constrained devices to access cloud data. This is demonstrated by our experimental results.

Future work includes evaluating our proposal in a real-world cloud deployment.

8. Acknowledgement

Jian Weng was supported by National Science Foundation of China under Grant No 61272413 and the Invitation Programs for Foreign-based Researchers provided by NICT.

References

[1] M. Blaze, G. Bleumer, M. Strauss, Divertible protocols and atomic proxy cryptography, in: Advances in Cryptology-Eurocrypt'98, LNCS 1403, 1998, pp. 127–144. doi:10.1007/BFb0054122.

[2] Q. Tang, Type-based proxy re-encryption and its construction, in: Indocrypt'08, 2008, pp. 130–144. doi:10.1007/978-3-540-89754-5_11.

[3] J. Weng, R. H. Deng, X. Ding, C.-K. Chu, J. Lai, Conditional proxy re-encryption secure against chosen-ciphertext attack, in: ASIACCS'09, 2009, pp. 322–332. doi:10.1145/1533057.1533100.

[4] Cloud security alliance: Security guidance for critical areas of focus in cloud computing, http://www.cloudsecurityalliance.org, 2009.

[5] European network and information security agency: Cloud computing risk assessment, november 2009, http://www.enisa.europa.eu/act/rm/_les/deliverables/cloud-computing-risk-assessment.

[6] Gartner: Don't trust cloud provider to protect your corporate assets, may 28, 2012, http://www.mis-asia.com/resource/cloud-computing/gartner-dont-trust-cloud-provider-to-protect-your-corporate-assets.

[7] J. Weng, Y. Yang, Q. Tang, R. H. Deng, F. Bao, Efficient conditional proxy re-encryption with chosen-ciphertext security, in: 12th International Conference on Information Security, ISC'09, LNCS 5735, 2009, pp. 151–166. doi:10.1007/978-3-642-04474-8_13.

[8] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, in: IEEE Symposium on Security & Privacy, S&P'07, 2007, pp. 321–334. doi:10.1109/SP.2007.11.

[9] V. Goyal, O. Pandy, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: ACM Computer and Communications Security Conference, CCS'06, 2006, pp. 89–98. doi:10.1145/1180405.1180418.

[10] Y. Yang, H. Lu, J. Weng, Y. Zhang, K. Sakurai, Fine-grained conditional proxy re-encryption and application, in: 8th International Conference on Provable Security, ProvSec'14, LNCS 8782, 2014, pp. 206–222. doi:10.1007/978-3-319-12475-9_15.

[11] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, ACM Transactions on Information and System Security (TISSEC) 9 (1) (2006) 1–30. doi:10.1145/1127345.1127346.

[12] R. Canetti, S. Hohenberger, Chosen-ciphertext secure proxy re-encryption, in: ACM CCS 2007, 2007, pp. 185–194. doi:10.1145/1315245.1315269.

[13] B. Libert, D. Vergnaud, Unidirectional chosen-ciphertext secure proxy re-encryption, in: PKC'08, LNCS 4929, 2008, pp. 360–379. doi:10.1007/978-3-540-78440-1_21.

[14] R. Canetti, H. Krawczyk, J. B. Nielsen, Relaxing chosen-ciphertext security, in: Advances in Cryptology-Crypto'03, LNCS 2729, 2003, pp. 565–582. doi:10.1007/978-3-540-45146-4_33.

[15] R. H. Deng, J. Weng, S. Liu, K. Chen, Chosen-ciphertext secure proxy re-encryption without pairings, in: CANS 2008, 2008, pp. 1–17. doi:10.1007/978-3-540-89641-8_1.

[16] J. Shao, Z. Cao, Cca-secure proxy re-encryption without pairings, in: PKC'09, LNCS 5443, 2009, pp. 357–376. doi:10.1007/978-3-642-00468-1_20.

[17] J. Weng, S. S. M. Chow, Y. Yang, R. Deng, Efficient unidirectional proxy re-encryption, iacr eprint: eprint.iacr.org/2009/189.pdf, 2009.

[18] X. Liang, Z. Cao., H. Lin, J. Shao, Attribute-based proxy ee-encrytpion with delegating capabilities, in: ACM ASIACCS'09, 2009, pp. 276–286. doi:10.1145/1533057.1533094.

[19] S. Luo, J. Hu, Z. Chen, Ciphertext policy attribute-based proxy re-encryption, in: International Conference on Information and Communications Security (ICICS'10), 2010, pp. 401–415. doi:10.1007/978-3-642-17650-0_28.

[20] K. Liang, et. al., A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing, Future Generation Computer Systemsdoi:doi:10.1016/j.future.2014.11.016.

[21] K. Liang, L. Fang, D. S. Wong, W. Susilo, A ciphertext-policy attribute-based proxy re-encryption scheme for data sharing in public clouds, Concurrency and Computation: Practice and Experience 27 (8) (2015) 2004–2027. doi:DOI:10.1002/cpe.3397.

[22] Y. Kawai, Outsourcing the re-encryption key generation: Flexible ciphertext-policy attribute-based proxy re-encryption, in: International Conference on Information Security Practice and Experience (ISPEC'15), 2015, pp. 301–315. doi:10.1007/978-3-319-17533-1_21.

[23] A. Sahai, H. Seyalioglu, B. Waters, Dynamic credentials and ciphertext delegation for attribute-based encryption, in: Crypto'12, LNCS 7417, 2012, pp. 199–217. doi:10.1007/978-3-642-32009-5_13.

[24] J. Zhao, D. Feng, Z. Zhang, Attribute-based conditional proxy re-encryption with chosen-ciphertext security, in: IEEE GLOBECOM 2010, 2010, pp. 1–6. doi:10.1109/GLOCOM.2010.5684045.

[25] R. Canetti, S. Halevi, J. Katz, Chosen-ciphertext security from identity-based encryption, in: Advances in Cryptology-Eurocrypt'04, LNCS 3027, 2004, pp. 207–222. doi:10.1007/978-3-540-24676-3_13.

[26] E. Fujisaki, T. Okamoto, Secure integration of asymmetric and symmetric encryption schemes, in: In Advances in Cryptology-Crypto'09, LNCS 1666, 1999, pp. 80–101. doi:10.1007/s00145-011-9114-1.

[27] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang, Y. Zhao, Generic construction of chosen ciphertext secure proxy re-encryption, in: CT-RSA'12, 2012, pp. 349–364. doi:10.1007/978-3-642-27954-6_22.

[28] V. Shoup, Lower bounds for discrete logarithms and related problems, in: Advances in Cryptology-Eurocrypt'97, 1997, pp. 256–266. doi:10.1007/3-540-69053-0_18.

[29] J. Schwartz, Fast probabilistic algorithms for verification of polynomial identities, Journal of ACM 27 (4) (1980) 701–717. doi:10.1145/322217.322225.

[30] R. Zippel, Probabilistic algorithms for sparse polynomials, in: International Symposiumon on Symbolic and Algebraic Computation, LNCS 72, 1979, pp. 216–226.

[31] B. Waters, Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, in: PKC'11, 2011, pp. 53–70. doi:10.1007/978-3-642-19379-8_4.

[32] R. Ostrovsky, A. Sahai, B. Waters, Attribute-based encryption with non-monotonic access structures, in: ACM CCS'07, 2007, pp. 195–203. doi:10.1145/1315245.1315270.

[33] J. Katz, A. Sahai, B. Waters, Predicate encryption supporting disjunctions, polynomial equations, and inner products, in: Advances in Cryptology-Eurocrypt'08, LNCS 4965, 2008, pp. 146–162. doi:10.1007/s00145-012-9119-4.

[34] D. Boneh, A. Sahai, B. Waters, Functional encryption: Definitions and challenges, in: Theory of Cryptography Conference, TCC'11, 2011, pp. 253–273. doi:10.1007/978-3-642-19571-6_16.

[35] A. Lewko, T. Okamoto, A. Sahai, T. Takashima, B. Waters, Fully secure funtional encryption: Attibute-based encryption and (hierarchial) inner product encryption, in: Advances in Cryptology-Eurocrypt'10, LNCS 6110, 2010, pp. 62–91. doi:10.1007/978-3-642-13190-5_4.

[36] N. Attrapadung, H. Imai, Attribute-based encryption supporting direct/indirect revocation modes, in: IMA International Conference on Cryptography and Coding, 2009, pp. 278–300. doi:10.1007/978-3-642-10868-6_17.

[37] G. Wang, Q. Liu, J. Wu, Hierarhical attribute-based encryption for fine-grained access control in cloud storage services, in: ACM CCS'10, 2010, pp. 735–737. doi:10.1145/1866307.1866414.

[38] S. Yu, C. Wang, K. Ren, W. Lou, Achieving secure, scalable, and fine-grained data access control in cloud computing, in: IEEE INFOCOM'10, 2010, pp. 1–9. doi:10.1109/INFCOM.2010.5462174.

[39] D. Boneh, X. Ding, G. Tsudik, C. Wong, A method for fast revocation of public key certificates and security capabilities, in: USENIX Security, 2001.

[40] M. Green, S. Hohenberger, B. Waters, Outsourcing the decryption of abe ciphertexts, in: USENIX Security'11, 2011.

[41] Y. Chen, L. Jiang, S. Yiu, M. Au, X. Wang, Fully-rcca-cca-secure ciphertext-policy attribute based encryption with security mediator, in: 16th International Conference on Information and Communications Security, ICICS'14 (to appear), 2014.