Conditional Proxy Re-Encryption for Secure Big Data Group Sharing in Cloud Environment

Junggab Son*, Donghyun Kim†, Rasheed Hussain*, and Heekuck Oh*‡
* Department of Computer Science and Engineering, Hanyang University, ERICA Campus, South Korea
E-mail: jgson@infosec.hanyang.ac.kr, rasheed@hanyang.ac.kr, hkoh@hanyang.ac.kr
† Department of Mathematics and Physics, North Carolina Central University, Durham, NC 27707, USA
E-mail: donghyun.kim@nccu.edu
‡ Corresponding Author.

***Abstract*—Conditional PRE (CPRE) is a novel public key primitive which enables the group sharing of confidential data without revealing its plaintext or decryption key to outside the group member. Previously, several efforts are made to facilitate CPRE in group data sharing in cloud environment. The main drawback of the state-of-art CPRE schemes for this purpose is that whenever the group member changes, the originator of the data needs to download all of the existing data on the cloud, encrypt them again with a new condition value, and uploads them to the cloud. As a result, they are not suitable for secure big data sharing among group member in cloud environment. In this paper, we introduce a new CPRE called the outsourcing CPRE scheme (O-CPRE) which reduces the client overhead drastically. When the membership of the group changes, in O-CPRE, the originator only needs to select a new condition value and upload it to the cloud. In addition, O-CPRE will move a part of client overhead at the initial setup stage and at the decryption of each message from the client to the cloud. As a result, O-CPRE is much more suitable for secure big data sharing in cloud environment than the other existing schemes.**

***Index Terms*—Cloud computing, cloud storage, big data, proxy re-encryption, conditional proxy re-encryption, efficiency.**

I. Introduction

The recent advances in cloud computing have dramatically altered the shape of current information technology industry. This new computing paradigm enables companies to purchase the pay-per-use computing resources in need from a cloud service provider, which is typically cheaper than establishing their own computing environment [1]. However, to make this innovative model more viable, it is crucial to guarantee the security of the data of the companies in the public cloud from any type of adversary including the cloud provider [2]–[4]. To preserve the confidentiality of the data stored at a cloud storage server, one can encrypt the data using its secure key before sending it to the server and decrypt it after downloading from the server. However, this strategy is extremely inefficient in cloud environment due to the heavy overhead at the user. Meanwhile, it is also not appropriate for a cloud server to obtain the encryption/decryption privileges on behalf of the user due to the confidentiality issue. This conundrum of designing efficient and secure cloud storage becomes even complicated if the data is being shared among a group of users whose membership is not necessarily static.

To address the issue of secure group data sharing at a cloud environment, the concept of *proxy re-encryption (PRE)* has been introduced in the literature [5]–[7], where a data originator u_i delegates the privilege of re-encrypting a data m encrypted by u_i for another user u_j to a cloud storage such that (a) the cloud cannot see the plaintext of m using the privilege, but (b) u_j can recover the plaintext from the re-encrypted message (by the cloud) with its own private key. This strategy elegantly moves the user overhead to the cloud environment, but the existing PRE schemes suffer from one common security concern called the abuse of re-encryption. In detail, once u_i sends a re-encryption (privilege) key to the cloud storage for a new group member u_j, u_j can collude with the cloud storage operator to decrypt all of the old and messages originated from u_i for the other group members even though u_j is not authorized to access them.

To deal with this issue, the *conditional proxy re-encryption (CPRE)* has been introduced [8], [9]. In CPRE, u_i creates encrypted messages as well as re-encryption keys (one for each member of a group) with a certain condition value (for the whole group). In detail, given a group, u_i first selects a condition value w. Then, use w and its own private key to encrypt the messages to be shared. At the same time, u_i create a re-encryption key for each member of the group using w and the public key of the member. Then, the encrypted messages, w, and the re-encryption keys are sent to the cloud storage for message re-encryption and distribution like PRE. Once the membership of the group is changed, u_i selects a new condition value w' and repeat the process for the new messages generated after the group change as well as the existing messages in the cloud storage. In this way, any revoked member left the group cannot access new messages, and any new member cannot access old messages.

In [15], Son et al. pointed out that the existing CPRE can suffer from a serious efficiency issue if the size of the group is large and the membership of the group changes frequently. This is because whenever membership changes, the re-encryption key for each group member is newly generated by each data originator and the existing messages in the cloud have to be encrypted again using this new condition value. To alleviate the overhead, they proposed an *efficient CPRE (E-CPRE)* in which whenever the membership of the group changes, a data originator needs to select a new condition value and sends it to the cloud server, but does not need to create and upload the re-encryption keys for the other group members. When the group membership changes, E-CPRE works more efficiently than the existing CPRE since it reduces the overhead of the data originator to compute new re-encryption keys and upload them to the cloud. However, both CPRE and E-CPRE require to encrypt the existing messages on the cloud using the new condition value and upload them. As a result, they are inefficient for secure big data group sharing in cloud environment.

In this paper, we introduce an improved version of E-CPRE, namely the outsourcing CPRE scheme (O-CPRE). Unlike the other existing CPRE schemes including E-CPRE, in O-CPRE, the only tasks required for the originator of messages to perform when the group membership is changed are

(a) computing a condition value changing key (CCK), which includes a new condition value, and
(b) sending the CCK to the cloud storage. Then, the cloud storage will use the CCK to transform existing ciphertexts such that the messages inside the ciphertexts are encrypted using the new conditional value.

Clearly, the overhead of the message originator under the group membership change is drastically reduced. In addition, the advantage of O-CPRE over E-CPRE (and the other existing CPRE schemes) is getting magnified as the amount of the data in the cloud storage increases. As a result, O-CPRE is much more desirable than the other existing CPRE schemes for secure group big data group sharing in cloud environment.

The rest of this paper is organized as follows. Section II discusses our system and security models. In Section III, we introduce some preliminaries. Our main contribution, the new conditional proxy re-encryption scheme with much lower user overhead, is presented in Section IV. The security and efficiency analysis of the proposed scheme is in Section V. Finally, we conclude this paper in Section VI.

II. System Model and Security Model

A. System Model

Fig. I. illustrates a system model for data sharing in cloud computing. We define three different entities and they can be identified as follows.

- **Outsourcing Server**: This entity has important role in our system model. Outsourcing Server is to complete delegated expensive operations to overcome the disadvantage that the re-encryption key generation and decryption phase in typical CPRE requires a lot of overload operations at client. In addition, we assume a honest-but-curious system model. The outsourcing server in this model runs a given protocol as best as it can and pays attention to client's data simultaneously. Therefore, there is possibility of passive attacks such as looking at content of data or eavesdropping in data transfer process. Since the cloud server can take on a role of outsourcing server, in the following descriptions we will use outsourcing server and cloud server interchangeably.

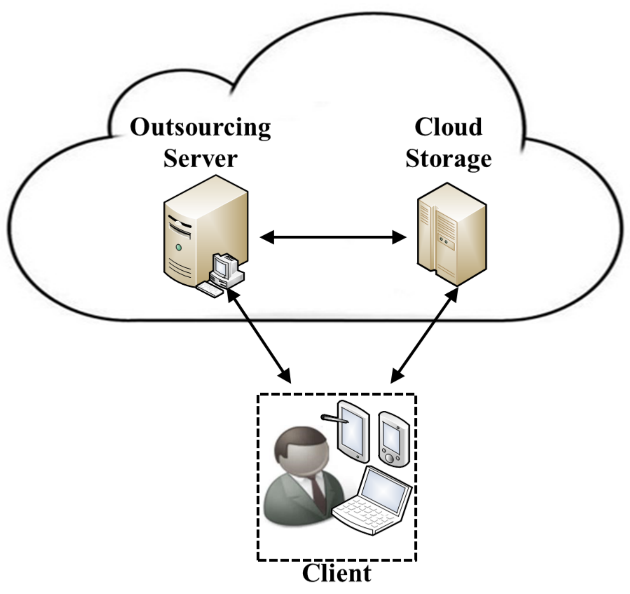


Fig. 1: System Model.

- **Cloud Storage**: It provides the users with computing and storage resources in pay-per-use basis. The cloud storage maintains the users' ciphertexts, re-encrypts, and sends data on the request of a customer. In the following descriptions, we will use "Cloud Storage" and "Cloud" interchangeably.
- **Client**: This entity has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation. Clients can be either individual consumers or organizations. The client can connect to the cloud with various devices. It includes resource-constrained mobile devices such as smart phone or tablet. Therefore, it can be relieved of the burden of maintaining and computation by storing the large data files in the cloud storage. In our environment, a client can be the originator of data files as well as destination of data sharing.

B. Security Model

We argue that the data sharing scheme is secure when it satisfies the following conditions.

(a) No polynomial time extractor exists that can recover the original data files by carrying out partial decryptions. This condition is applied to the outsourcing server.
(b) No polynomial time extractor exists that can generate re-encryption key by carrying out partial re-encryption key computation. This condition is applied to the outsourcing server.
(c) No polynomial time extractor exists that can recover the condition value or insert another condition value to the ciphertext. This condition is applied to the cloud.

III. Preliminaries

A. Notations

The notation used in this paper are listed in Table 1.

TABLE I: Notations.

Notation	Description
q	k-bit prime number
$\mathbb{Z}_q$	Integers modulo q
$\mathbb{G}, \mathbb{G}_T$	Cyclic group with prime order q
g	Generator of $\mathbb{G}$
e	Bilinear pairing that satisfies with $\mathbb{G} \times \mathbb{G}_T$
U_i	Client i
pk_i, sk_i	Public/private key pair of U_i
n	Polynomial in k
m	Data, $m = \{0,1\}^n$
w	Condition value
r	Random number that is included in $\mathbb{G}$
s	Random number that is included in $\mathbb{Z}_q^*$
CT_i	Ciphertext generated by U_i
RK_{part_j}	Partial re-encryption key for j
$RK_{(i \to j)}$	Re-encryption key for $U_i \to U_j$
CCK	Condition value Changing Key
H_1	$\{0,1\}^* \to \mathbb{Z}_q$
H_2	$\{0,1\}^* \to \mathbb{G}$
H_3	$\mathbb{G} \to \mathbb{Z}_q$

B. Assumption

In this paper, we establish following two assumptions.

- **Assumption 1**: The proposed scheme assumes that there is no collusion between a client and the cloud. A client does not send and receive information that is necessary to decrypt data to/from the cloud.
- **Assumption 2**: The proposed scheme assumes that a client does not share its condition values with other clients.

C. Cryptographic Background

Next, we introduce two important definitions.

Definition 1 (Bilinear map). *A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties [11].*

(a) Computable: there exists an efficiently computable algorithm for computing e,

(b) Bilinear: for all $h_1, h_2 \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p, e(h_1^a, h_2^b) = e(h_1, h_2)^{ab}$, and

(c) Nondegenerate: $e(g, g) \neq 1$, where g is a generator of $\mathbb{G}$.

Definition 2 (DBDH). *The Decisional Bilinear Diffie-Hellman (DBDH) problem in groups $(\mathbb{G}, \mathbb{G}_T)$ is, given a tuple $(g, g^a, g^b, g^c, Z) \in \mathbb{G}^4 \times \mathbb{G}_T$ with unknown $a, b, c \in_R \mathbb{Z}_q$, whether $Z = e(g, g)^{abc}$. A polynomial-time algorithm $\mathcal{B}$ has advantage ϵ in solving the DBDH problem in groups $(\mathbb{G}, \mathbb{G}_T)$, if*

$$|(Pr[(g, g^a, g^b, g^c, \mathbb{Z} = e(g, g)^{abc}) = 1] \\ - Pr[(g, g^a, g^b, b^c, \mathbb{Z} = e(g, g)^d) = 1])| \geq \epsilon,$$

where the probability is taken over the random choices of $a, b, c, d \in \mathbb{Z}_q$, the random choice of g in $\mathbb{G}$, and random bits consumed by $\mathcal{B}$.

IV. Proposed Securely Outsourcing CPRE

When a condition value needs to be changed, the originator has to generate new condition value. Then, the originator generates CCK which contains new condition value and sends it to cloud. Note that CCK is generated through exponential operations and thus the condition value is protected by discrete logarithm problem. Once the cloud storage receives CCK, it performs pairing operations with CCK and multiplies it with stored ciphertexts to change old condition value. Since the cloud can transform all the ciphertexts that include old condition value using CCK, the originator does not need to manually retrieve the ciphertexts to change condition value like the other existing CPRE schemes.

Additionally, O-CPRE partially delegates the operations which are traditionally burdened to the clients in the existing CPRE to the cloud and further reduce the overhead of client in two ways. First, our scheme delegates re-encryption key generation process. In detail, we introduce a new two-phase re-encryption strategy. The first phase is performed by the cloud and the second phase is performed by the originator after receiving the result from the first phase. Although a re-encryption key generation process is performed only once at the initial setup of the system, our scheme can significantly reduce a burden of the clients by delegating exponential computation to the cloud. Second, our scheme moves a part of the decryption process from the clients to the cloud. That is, per the client's request of partial decryption, a cloud server performs pairing operation with the public key of the client and a part of the re-encryption key for the client. After receiving result of the partial decryption from the cloud, the client can complete the decryption with a single exponential operation.

A. Setup

On input a security parameter 1^k, the setup process first determines $(q, \mathbb{G}, \mathbb{G}_T, e)$. Next, the cloud chooses $g \in_R \mathbb{G}$, and five hash functions H_1, H_2, H_3, H_4, and H_5. The global parameter is

$$((q, \mathbb{G}, \mathbb{G}_T, e), g, n, H_1, H_2, H_3).$$

The client generates a public/private key pair (pk_i, sk_i). The client picks $x_i \in_R \mathbb{Z}_q$, and computes $g^{(x_i)}$. The private key is $sk_i = x_i$ and the public key is $pk_i = g^{(x_i)}$.

B. Re-encryption Key Generation

To outsource re-encryption key generation process, we construct that re-encryption key can be generated through two diffrent phases. First, Outsourcing server computes partial computation that needs heavy exponential operation. Then the client generates re-encryption key through one multiplication. Using this method, computational overhead of mobile client can reduce drastically.

The U_i who is the originator of the ciphertext, can generate the re-encryption key as meaning of allowing U_j to share the data. A client requests partial re-encryption computation with issuing $g^{s \cdot sk_i}$, then outsourcing server computes following partial re-encryption key:

$$RK_{part\_j} = pk_j^{H_3(g^{s \cdot sk_i})}.$$

On receiving this, The U_i completes the re-encryption key as

$$\begin{aligned} RK_{(i\to j)} &= (rk_1, rk_2) \\ rk_1 &= RK_{part\_j} \cdot g^s \cdot r^{-sk_i} \\ &= r^{-sk_i} \cdot pk_j^{s+H_3(g^{s\cdot sk_i})}, \\ rk_2 &= g^{s\cdot sk_i}. \end{aligned}$$

C. *Data Encryption*

CPRE scheme has two encryption levels. The first level encryption generates a ciphertext which does not allow re-encryption. Therefore, the first level ciphertext is generated without condition value. And the second level encryption generates a ciphertext which allows re-encryption. The second level ciphertext includes condition value which is used to control decrypt permission. The client can chooses the encryption level depending on the importance of the data. The first level ciphertext can be generated by following process.

The U_i first picks $R \in_R \mathbb{G}$ and $s \in_R \mathbb{Z}_q^*$. The client computes $r = H_1(m, R)$, and generates the first level ciphertext $CT_i = (C_1, C_2, C_3, C_4)$ as

$$(g^r, R \cdot e(g, pk_i)^{-r\cdot s\cdot H_5(pk_i^s)}, m \oplus H_3(R), g^s).$$

For data sharing, the client generates the second level ciphertext by following process:

$$\begin{aligned} CT_i &= (C_1, C_2, C_3, C_4) \\ C_1 &= g^W, W = H_1(w) \\ C_2 &= e(pk_i, r)^W \\ C_3 &= m \cdot e(g, g)^s \\ C_4 &= H_2(C_1, C_2, C_3). \end{aligned}$$

Then, the client sends this encrypted data to the cloud.

D. *Data Re-encryption*

The cloud can re-encrypt the CT_i by U_j's request. Re-encryption process is as follow:

$$\begin{aligned} CT_j &= (\overline{C}_1, \overline{C}_2, \overline{C}_3, \overline{C}_4) \\ \overline{C}_1 &= C_1, \\ \overline{C}_2 &= C_2 \cdot e(C_1, rk_1) \\ &= e(pk_i, r)^W \cdot e(g^W, r^{-sk_i} \cdot pk_j^{s+H_3(g^{s\cdot sk_i})}) \\ &= e(g^W, pk_j^{s+H_3(g^{s\cdot sk_i})}), \\ \overline{C}_3 &= C_3 = m \cdot e(g, g)^s, \\ \overline{C}_4 &= rk_2. \end{aligned}$$

E. *Changing Condition Value*

In previous CPRE scheme, a client has to receive its own ciphertext and encrypt it again with new condition value to change condition value. It is cymbersome to client. Therefore, we design the system where cloud can change condition value by condition value changing key that issued by a client. To change condition value, the client performs following process with newly generated condition value, w':

$$\begin{aligned} W' &= H_1(w') \\ t &= W' - W. \end{aligned}$$

The client sends $g^{W'}$ and r^t to the cloud. Now, cloud generates new C_2 that has newly issued condition value by performing following process:

$$\begin{aligned} C'_1 &= g^{W'}, \\ C'_2 &= C_2 \cdot e(pk_i, r^t) \\ &= e(pk_i, r)^{W'}. \end{aligned}$$

F. *Partial Decryption*

Surely, a client can decrypt a ciphertext with whole decryption process as well as delegate to make outsourcing server computing partial decryption process. To outsource partial decryption, we use condition value as secret factor. Since outsourcing server cannot obtain condition value in our scheme, this secret factor makes partial decryption possible without revealing private key or original message. For a client's request of partial decryption, outsourcing server computes following process:

$$\begin{aligned} CT_{part} &= \frac{\overline{C}_2}{e(\overline{C}_1, pk_j^{H_3(rk_2)})} \\ &= \frac{e(g^W, pk_j^s) \cdot e(g^W, pk_j^{H_3(s\cdot sk_i)})}{e(g^W, pk_j^{H_3(s\cdot sk_i)})} \\ &= e(g^W, pk_j^s). \end{aligned}$$

The outsourcing server sends CT_{part} to the U_j.

G. *Data Decryption*

The U_i can decrypt the first level ciphertext by following process. First, the U_i computes

$$R = C_2 \cdot e(C_1, C_4)^{sk_i\cdot H_5(C_4^{sk_i})}, \text{ and } m = C_3 \oplus H_3(R).$$

Next, the U_i checks $g^{(H_1(m,R))} = C_1$ to confirm the validity of the data.

U_j can decrypt the partial decrypted ciphertext by following process. The U_j can obtain a plaintext only if it knows the condition value.

$$\begin{aligned} m &= \frac{\overline{C}_3}{CT_{part}^{1/(H_1(w)\cdot sk_j)}} \\ &= \frac{\overline{C}_3}{e(g^W, pk_j^s)^{1/(H_1(w)\cdot sk_j)}}. \end{aligned}$$

V. Analysis of Proposed Scheme

A. *Security*

This subsection analyzes the security of the method proposed on the security model in Section 4.2. The main purpose of the method proposed is to delegate heavy computational

TABLE II: Comparison of Computational Overhead in client side

	CPRE [8]	Efficient CPRE [15]	Proposed Scheme
Re-encryption key generation	$4t_e + 3t_m + 2t_h$	$3t_e + 2t_m + 1t_h$	$3t_e + 3t_m$
Decryption	$1t_p + 2t_e + 1t_m + 2t_h$	$1t_p + 2t_e + 3t_m + 2t_h$	$1t_e + 2t_m + 1t_h$
Changing a condition value	$1t_p + 4t_e + 4t_m + 3t_h$	$1t_p + 2t_e + 1t_m + 2t_h$	$2t_e + 1t_h$

overhead during data sharing process through CPRE. Therefore, we need to prove that the proposed scheme can securely delegate a computation of CPRE.

Theorem 1. *The proposed scheme is secure against chosen-ciphertext attack under DBDH assumption.*

Proof. For robustness of our scheme, we make use of the idea in [8], [10] by making relationship between sub re-encryption keys. The design of scheme proposed is based on n-Quotient Bilinear Diffie-Hellman Assumption, the scheme shows security by calculating complexity of DBDH. The re-encryption key consists of two values, (rk_1, rk_2) and by including $H_3(pk_B^{rk_2}) = H_3(pk_B^{g^{s \cdot sk_i}})$ to the rk_1, this makes relationship between rk_1 and rk_2. In this way, the proposed scheme can have robustness for the chosen ciphertext attack. In [8], they already proved the security against chosen-ciphertext attack, we do not repeat verification in detail. □

Theorem 2. *An outsourcing server cannot recover a plaintext by carrying out multiple partial re-encryptions.*

Proof. To outsource decryption securely, we use a condition value as blinding factor which is the shared secret among the group of users. Through this blinding factor, our scheme is able to delegate partial decryption operation to outsourcing server without private key or plaintext leakage. Additionally, we argue that the condition value is hard to recover under the DBDH assumption. □

Theorem 3. *An outsourcing server cannot generate a new re-encryption key by carrying out multiple partial re-ecnryption key computations.*

Proof. A first re-encryption key rk_1 in our scheme consists of two part. First part is computed by outsourcing server and second part is computed by client. A client can complete a re-encryption key by merging two parts. Here, first part of rk_1 has only public value. The outsourcing server adds a hash value to the robustness of CPRE. Although the outsourcing server has a lot of this public values and re-encryption keys, it needs to know additional data to generates a new re-encryption key. The outsourcing server has to know s or r and it is secure under difficulty of discrete logarithm problem [12]. Additionally, we assumed that honest-but-curious system of outsourcing server and outsourcing server sends validity value with RK_{part}. Therefore, we omit the proof about forging a RK_{part} that the outsourcing server generates it with another public key rather than requested public key. □

Theorem 4. *A cloud cannot recover the condition value or insert another condition value to the ciphertext by carrying out multiple condition value changing processes.*

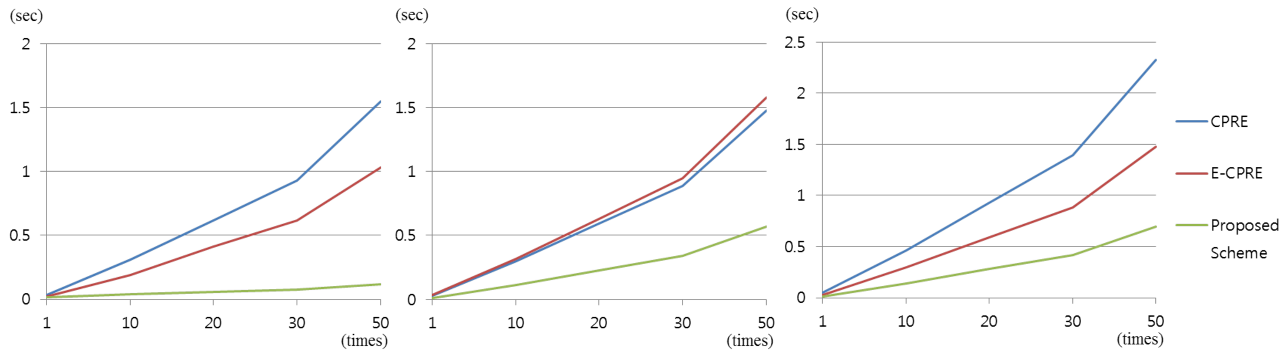
Proof. A client has to send $g^{W'}$ and $r^{W'-W}$ to change condition value. To forge a condition value, the cloud needs to compute r^{x-w} (where x is a condition value that the cloud wants to forge). But, it is computationally impossible to the cloud restoring W from g^W and computing r^{x-w} without having r. □

B. Efficiency

Our main design objective of O-CPRE is to delegate the heavy computational overhead on the client side (especially the data originator) to the cloud storage during data sharing in mobile cloud. The greatest advantage of our scheme is that the cost of expensive re-encryption key generation, decryption and condition value changing are significantly reduced by partially delegating the tasks to server. Apparently, O-CPRE performs much more eifficiently than the existing CPRE schemes from the client perspective. In Table. 2, we present the result of computational overhead comparison among CPRE, E-CPRE, and O-CPRE. In this table, t_p, t_e, t_m, t_h represent the computational cost of a bilinear pairing, an exponentiation, a multiplication, and a hash, respectively.

The computational overhead of re-encryption key generation process in proposed scheme seems similar to CPRE scheme. Since a client can reuse the part of re-encryption key which is used to complete a re-encryption key, a client has higher efficiency for generating more re-encryption keys in our scheme.

Fig. 2 shows an overall simulation results in re-encryption key generation, decryption and changing a condition value aspects.Our experiment is implemented on an Intel core (TM) i7 processor running at 1.5 GHZ, 8.00 GB of RAM, and SSD Serial ATA 3.0 Gbit/s drive with a 16MB buffer. All algorithms are implemented using C language in Linux Ubuntu 12.04 LTS 32 bit. Our code uses pairing based cryptography (PBC) library version 0.5.12. All experimental results represent the average of 10 trials. Fig. 2 (a) shows comparison in point of re-encryption key generation among CPRE schemes. Because of our scheme can delegate re-encryption key generation process partially, it can provide high efficiency in client side. Our scheme can delegate approximately 92.5% of computation for re-encryption key generation. Since originator needs to receive public key of other client to generate re-encryption key. In our scheme, originator receives cloud computed partial re-encryption key instead of public key. Therefore, overall communication overhead is similar as previous CPRE schemes. Fig. 2 (b) shows comparison in point of decryption among CPRE schemes. Using partial decryption function of our



(a) Re-encryption key generation

(b) Decryption

(c) Changing a condition value

Fig. 2: Simulation results

scheme, we can delegate approximately 61.7% of computation for decryption. Fig. 2 (c) shows comparison in point of changing condition values. Our scheme can delegate approximately 30% of computation by CCK. These delegation means less computation is needed to sharing data using CPRE for mobile clients.

VI. Conclusion

In this paper, we propose an outsourcing CPRE scheme which can delegate heavy computation securely during data sharing in mobile cloud environment. The proposed scheme introduces outsourcing server which performs delegated operations due to two characteristics. First, partial re-encryption key generation and decryption phases can be delegate to outsourcing server. Second, condition value changing phase can be delegate to cloud server with CCK which is new approach of our scheme. By delegating three phases of CPRE, our scheme can significantly reduce burden of a client.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education, Science and Technology (No. 2012-R1A2A2A01046986). This work was also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012-R1A1A2009152). This research was supported in part by the MSIP (Ministry of Science, ICT & Future Planning, Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2013-H0301-13-1002) supervised by the NIPA (National IT Industry Promotion Agency). This work was also supported in part by US National Science Foundation (NSF) CREST No. HRD-1345219 and by US Army Research Office (ARO) No. W911NF-0810510.

References

[1] K. M. Khan, and Q. Malluhi, "Establishing Trust in Cloud Computing," *IT Professional*, vol. 12, issue 5, pp. 20-27, Sept. 2010.
[2] H. Takabi, J. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," *IEEE Security & Privacy*, vol. 8, issue 6, pp. 24-31, 2010.
[3] L. M. Kaufman, "Data Security in the World of Cloud Computing," *IEEE Security & Privacy*, vol. 7, issue. 4, pp. 61-64, 2009.
[4] K. Ren, C. Wang, and Q. Wang, "Security Chanllenges for the Public Cloud," *IEEE Internet Computing*, vol. 16, issue. 1, pp. 59-73, 2012.
[5] M. Mambo and E. Okamoto, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertext," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (TFECCS)*, 1997.
[6] M. Blaze, G. Beumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," *Proceeding of International Conference on the Theory and Application of Cryptographic Techniques (Eurocrypt)*, pp. 127-144, 1998.
[7] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, issue 1, pp. 1-30, 2006.
[8] J. Weng, Y. Yang, Q. Tang, R. H. Deng, and F. Bao, "Efficient Conditional Proxy Re-encryption with Chosen-Ciphertext Security," *Proceedings of the 12th Information Security Conference (ISC)*, pp.151-166, 2009.
[9] C.-K. Chu, J. Weng, S.S.M. Chow, J. Zhou, and R.H. Deng, "Conditional Proxy Broadcast Re-Encryption," *Proceedings of the 14th Australasian Conference on Information Security and Privacy (ACISP)*, pp. 327-342, 2009.
[10] R. H. Deng, J. Weng, S. Liu, and K. Chen, "Chosen-Ciphertext Secure Proxy Re-encryption without Pairings," *Proceedings of the 7th International Conference on Cryptology and Network Security (CANS)*, pp.1-17, 2008.
[11] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," *Proceedings of the 7th International Conference on Theory and Application of Cryptology and Information Security (ASIACRYPT)*, pp. 514-532, 2001.
[12] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. 31. issue. 4, pp. 469-472, 1985.
[13] D. Cash, E. Kiltz, and V. Shoup, "The Twin Diffie-Hellman Problem and Applications," *Proceeding of International Conference on the Theory and Application of Cryptographic Techniques (Eurocrypt)*, 2008.
[14] J. Zhao, D. Feng, and Z. Zhang, "Attribute-Based Conditional Proxy Re-Encryption with Chosen-Ciphertext Security," *Proceeding of Global Telecommunications Conference (GLOBECOM)*, 2010.
[15] J. Son, H. Kim, D. Kim, and H. Oh, "On Secure Data Sharing in Cloud Environment," *Proceeding of International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2014.