# On Secure Data Sharing in Cloud Environment

Junggab Son
Department of Computer Science and Engineering
Hanyang University
Ansan, South Korea
jgson@infosec.hanyang.ac.kr

Hyunbum Kim
Department of Computer Science
Bethune-Cookman University
Daytona Beach, FL 32114, USA
kimh@cookman.edu

Donghyun Kim
Department of Mathematics and Physics
North Carolina Central Univ.
Durham, NC 27707, USA
donghyun.kim@nccu.edu

Heekuck Oh*
Department of Computer Science and Engineering
Hanyang University
Ansan, South Korea
hkoh@hanyang.ac.kr

## ABSTRACT

Over years, cloud computing has been rapidly changing the shape of modern computing environment. The problem of how to keep the confidentiality of user data against malicious entities including a cloud service provider has been recognized as a significant issue. This problem becomes even more complicated if a data is shared among multiple users. Recently, the idea of proxy re-encryption has been introduced to support secure data sharing among group members in cloud environment. However, in this scheme, a malicious user can collude with the server to decrypt unauthorized messages. The conditional proxy re-encryption (CPRE) aims to fix this problem by introducing a condition value into message encryption process and re-encryption key generation. We observe that CPRE becomes significantly inefficient when the membership of the group changes very actively and the size of the group is large since a new condition value is selected and re-encryption keys have to be generated for each user whenever the group membership is changed. This paper introduces a new CPRE in which the condition value is not associated with re-encryption keys. Whenever a group membership is changed, only a new condition value is distributed to the users via cloud server. As a result, the overhead of each user becomes significantly reduced at each membership change.

*Corresponding author


*IMCOM (ICUIMC)* '14, January 9–11, 2014, Siem Reap, Cambodia


## Categories and Subject Descriptors

H.3.5 [**Information Systems**]: Online information services - data sharing; E.3 [**Data**]: Data encryption - public key cryptosystems

## General Terms

Security

## Keywords

Cloud Computing, Data Sharing, PRE (Proxy Re-Encryption), CPRE (Conditional PRE).

## 1. INTRODUCTION

The recent advances in cloud computing technology have drastically changed the shape of current computing industry. By exploiting the innovative technology, companies are able to purchase the computing resources in need from a cloud service provider such as Amazon EC2 rather than establishing and maintaining their own computing environment, which is usually much more expensive than paying corresponding cloud services per use. The cost saved in this way can be invested to improve the core competitiveness and productivity of the companies [1]. Due to the benefit, cloud computing paradigm has been adopted to a wide range of businesses and organization recently.

For a company, the transition from its own private computing environment to a cloud computing environment means a huge financial advantage. At the same time, that means that all the data of the company even including confidential data is in the hand of another organization, which is a serious security concern [2]. For instance, the data stored in a remote cloud storage can be exposed when the server is attacked [3]. Furthermore, a recent report shows that data leakage and privacy infringement by a cloud service provider are also significant issues [4]. Consequently, in order to adopt

the cloud computing paradigm to a wider range of applications, it is very crucial for a cloud system managing personal information or company secret to be able to guarantee that it can preserve the confidentiality of the data even against itself.

To prevent data exposure when a server is attacked, one may consider encrypting a data using its own key before sending it to the server and decrypting it after getting it from the server. However, the encryption/decryption strategy of data at the user side is not appropriate in cloud environment since this does not properly utilize the cloud computing environment and pushing too much overhead to users. On the other hand, it is also not acceptable for a cloud server to obtain the encryption/decryption privileges on behalf of users since the administrators of the cloud server cannot be trusted. This already complicated problem becomes an even more complicated conundrum if the data is being shared among a group of users whose membership is not necessarily static.

Recently, the concept of proxy re-encryption has been proposed to allow a group of users to share a confidential data at a remote server without exposing the data to the server [5]. In essence, a proxy re-encryption transform a ciphertext $C$, which is encrypted by the public key of a user $\mathcal{A}$, to another ciphertext $C'$ in a way that $C'$ can be decrypted by the private key of another user $\mathcal{B}$ [6, 7]. To make this possible, $\mathcal{A}$ needs to create a re-encryption key for $\mathcal{B}$ and sent this to the server along with $C$. Once $C$ is requested by $\mathcal{B}$, the server re-encrypts the $C$ and transform it into $C'$ using the re-encryption key from $\mathcal{A}$, and sends $C'$ to $\mathcal{B}$. During this process, the server (as well as anyone who broke into the server) is not able to see the plaintext of $C$. On the other hand, $\mathcal{B}$ will be able to decrypt $C'$ using its own private key. While the idea of re-encryption is appealing, there is one issue. Once $\mathcal{A}$ creates a re-encryption key for $\mathcal{B}$, $\mathcal{B}$ can collude with the server to decrypt all of the old and new messages $\mathcal{A}$ created/will create, to which $\mathcal{B}$ does not obtain the permission to access.

To solve this problem, the concept of *conditional proxy re-encryption (CPRE)* has been emerged [8, 9]. In this scheme, $\mathcal{A}$ creates an encrypted message along with re-encryption keys (one for each member of a group) with a certain condition value (for the whole group). The condition value is used when $\mathcal{A}$ encrypts a message as well as when it creates an re-encryption key for another user. As a result, a malicious user who can collude with the server's administrator still cannot decrypt messages encrypted by $\mathcal{A}$ as long as $\mathcal{A}$ does not provide an re-encryption key which contains the condition value associated with the messages.

**Contribution of This paper.** In this paper, we introduce a new conditional proxy re-encryption scheme which works more efficiently for the group data sharing in cloud environment especially when the size of the group is large and the membership of the group is dynamic. We observe that in the existing CPRE schemes, whenever the membership of the group is changed, the condition value should be changed and the re-encryption key should be generated for each data shared by the updated group. This means that the current CPRE schemes are not efficient to use in large size dynamic organization such as a large company. To deal with this issue, we design a new CPRE scheme in which the re-encryption key and the condition is separated. As a result, whenever the group membership is changed, only the condition value is changed and broadcasted to the rest of the members via the cloud environment. As a result, the massive overhead of producing re-encryption keys can be significantly mitigated.

The rest of this paper is organized as follows. Section 2 discusses related work. Some preliminaries are given in Section 3. Our main contribution, a new conditional proxy re-encryption scheme with lower user side overhead is presented in Section 4. The security and efficiency analysis of the proposed scheme is in Section 5. Finally, we conclude this paper in Section 6.

## 2. RELATED WORK

Alice stores a message to the cloud after encrypting it with its own keys. When Alice receives data sharing request from Bob, it encrypts the received message with Bob's private keys before sending it. However, this method does not utilize advantages of the cloud computing as Alice takes all burdens to be generated from sharing and sharing is not established when Alice is absent. To deal with this problem, a group key scheme can be applied. This method allows clients joining sharing to use the same keys for data encryption. Although this method is more effective than aforementioned method, there is a key renewal issue appeared according to adding or removing a sharing member.

Ateniese et al. [5] proposed proxy re-encryptions to secure distributed storage. In this scheme, the originator of data encrypts data with symmetric data keys, and encrypted it with master public key. If users want to access the data, the server uses proxy re-encryption to directly re-encrypt the appropriate data key from the master public key to a granted user's public key. However, there is a side effect. If Alice creates re-encryption key for Bob, all messages encrypted by Alice's private key including previously created and future ciphertext can be re-encrypted by the re-encryption key. We called this side effect as abuse of re-encryption key. And this scheme has security flaw of collusion attack between the untrusted server and any revoked malicious user can be launched, which enables them to learn the decryption keys of all the encrypted blocks.

ABE (Attribute based encryption). ABE that uses an attribute as a encryption factor allows only a user who possesses a secret key for an attribute to decrypt a ciphertext. In this method, Alice allocates an attribute value to each user, creates attribute policy and generates ciphertext that reflects the policy. This method has an advantage that Alice can classify sharing object in details based on attribute values. However, it could not utilize operation capability of the cloud. As a result, there is no room to improve performance of the system as a whole. In addition, attribute policy should be renewed as sharing members are added or removed.

Lu et al. [10] proposed a secure provenance scheme,

which is built upon group signatures and CP-ABE (Ciphertext Policy-Attribute Based Encryption). The system is set with a single attribute and each user obtains two keys after the registration: a group signature key and an attribute key. Thus, any user is able to encrypt a data file using ABE and others in the group can decrypt the data using their attribute keys. Meanwhile, the user signs encrypted data with her group signature key for privacy preserving and traceability. Therefore, their scheme hard to support user revocation.

Yu et al. [11] proposed a scalable and fine-grained data access control scheme in cloud computing based on the KP-ABE (Key Policy Attribute Based Encryption). The data originator uses a random data key to encrypt a data and the data key is encrypted with a set of attributes using KP-ABE. Then, the group manager assigns an access policy and the corresponding secret key to authorized users, such that a user can only decrypt a ciphertext if and only if the data file attributes satisfy the access policy. User revocation is open problem in ABE. To achieve user revocation, the manager delegates tasks of data file re-encryption and user secret key update to cloud servers. However, ABE is not suitable for cloud computing because it has much computational overhead to the client and cannot benefit from the cloud's resource.

# 3. PRELIMINARIES

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 condition values that are possessed by it with other clients.

Next, we introduce two important definitions.

DEFINITION 1 (BILINEAR MAP). *A bilinear map is a map* $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ *with the following properties [13].*

*(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* $\epsilon$ *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* $\mathcal{B}$*.*

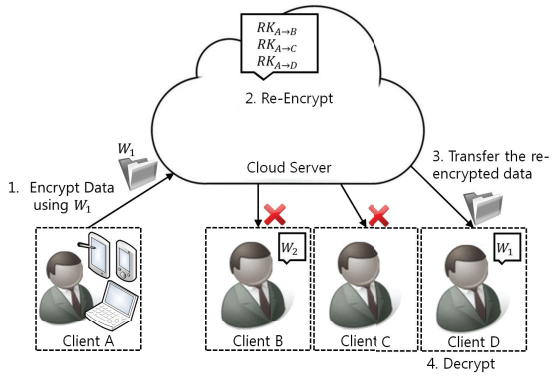


**Figure 1: System Model.**

## 3.1 System Model

Fig. 1. illustrates a system model for data sharing in cloud computing. We define two different entities and they can be identified as follows.

- **Cloud Server**: It provides users with operation and storage while users pay reasonable fee for the service. The cloud server stores user's ciphertext and re-encrypts and sends data on the request of a customer. This study proposes a honest-but-curious system model. The cloud server in this model runs a given protocol as best as it can and pays attention to user's data simultaneously. Therefore, there is possibility of passive attacks such as looking at content of data or eavesdropping in data transfer process. In addition, attacks to the CPRE scheme are also possible, such as comparison of findings from iterative operation of ciphertext or creating re-encryption keys to be sent to the cloud server. In the following descriptions, we will use Cloud Server, 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-constrict 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, client can be originator of data files as well as destination of data sharing.

## 3.2 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 multiple re-encryption. This condition is applied to the cloud server. The cloud server may make an attack by comparing results that are created by iterative re-encryption of a message and inferring a plain text

from the comparison. This condition includes security of re-encryption keys as the cloud possesses both ciphertexts and re-encryption keys.

(b) No polynomial time extractor exists that can recover the original data files without condition values. This condition is applied to a client. A client should not decrypt ciphertexts even if it receives them unless it knows their condition value. Similarly, a client should not find a condition value, on the basis of a ciphertext with the same condition value.

(c) No polynomial time extractor exists that can recover the original data files with only condition values. This condition is applied to the cloud. The cloud may receive a condition value with consent of a client that has a decoding authority. In this case, the cloud simply holds the condition value and messages and should not restore a plain text from them.

## 3.3 Design Goals

In addition to the security model introduced in Section 3.2, the proposed method is designed under the consideration of following properties.

- **Client Efficiency**: The proposed method aims to improve the client side efficiency, which is one of main motivation of cloud computing. As a result, the proposed strategy is designed to delegate overhead to the cloud server side as much as possible.
- **Managing Group Efficiency**: The method proposed is designed to manage the groups sharing data efficiently including the creation of a group as well as the addition and removal of a client to/from the group.

# 4. A NEW CONDITIONAL PROXY RE-ENCRYPTION WITH LOWER USER OVERHEAD

## 4.1 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

$$((q, \mathbb{G}, \mathbb{G}_T, e), g, n, H_1, H_2, H_3, H_4, H_5).$$

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

## 4.2 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 allow re-encryption. The second level ciphertext include condition value which used for control decrypt permission. The client can chooses the encryption level depending on the importance of the data. The first level ciphertext can generate by following process.

The $U_i$ first picks $R \in_R \mathbb{G}_T$ 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.

$$CT_i = (C_1, C_2, C_3, C_4) = (g^r, R \cdot e(pk_i, H_2(pk_i||w))^r, \\ m \oplus H_3(R), H_4(C_1, C_2, C_3, C_4)^r).$$

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

## 4.3 Data Re-encryption for Sharing

The $U_i$ who is originator of the ciphertext, can generate the re-encryption key as meaning of allow $U_j$ to share the data. The $U_i$ first picks $s \in_R \mathbb{Z}_q$, and generates the re-encryption key as

$$RK_{i \to j} = (rk_1, rk_2) = ((pk_j^{s \cdot H_5(pk_j^{s \cdot sk_i})})^{-sk_i}, pk_i^s).$$

Since the re-encryption key does not have a condition value, $U_i$ need to generates only one re-encryption key for $U_j$. Now, the cloud can re-encrypt the $CT_i$ by $U_j$'s request. Re-encryption process is as follow:

$$CT_i = (C_1, C_2, C_3, C_4)$$

and

$$CT_j = (\overline{C}_1, \overline{C}_2, \overline{C}_3, \overline{C}_4) \\ = (\overline{C}_1 = C_1, \overline{C}_2 = C_2 \cdot e(C_1, rk_1), \overline{C}_3 = C_3, \overline{C}_4 = rk_2) \\ = (g^r, R \cdot e(g, pk_j^{s \cdot sk_i \cdot H_5(pk_i^{s \cdot sk_i})}), m \oplus H_3(R), g^{s \cdot sk_i}).$$

## 4.4 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. The $U_j$ can decrypt the re-encrypted ciphertext by following process. The $U_j$ can obtain a plaintext only if knows the condition value. First, the $U_j$ computes

$$R = \frac{\overline{C}_2}{e(\overline{C}_1, H_2(w)) \cdot e(\overline{C}_1, \overline{C}_4^{sk_j \cdot H_5(\overline{C}_4^{sk_j})})}$$

and

$$m = \overline{C}_3 \oplus H_3(R).$$

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

## 4.5 Changing a Condition Value

To securely share data using our proposed scheme, the originator of data has to transfer a condition value to the clients who want to access the data. The originator transmit the condition value when the protocol is

**Table 1: Notations.**

| Notation | Description |
|---|---|
| $q$ | $k$-bit prime number |
| $\mathbb{Z}_q$ | Integers modulo $q$ |
| $\mathbb{G}, \mathbb{G}_T$ | Cyclic group with prime order $q$ |
| $e$ | 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 |
| $CT_i$ | Ciphertext generated by $U_i$ |
| $RK_{(i \to j)}$ | Re-encryption key for $U_i \to U_j$ |
| $g, r, s$ | Random number that is included in $\mathbb{G}, \mathbb{G}_T, \mathbb{Z}_q$, respectively |
| $H_1$ | $\{0,1\}^* \to \mathbb{Z}_q$ |
| $H_2$ | $\{0,1\}^* \to \mathbb{G}$ |
| $H_3$ | $\mathbb{G} \to \{0,1\}^n$ |
| $H_4$ | $\{0,1\}^* \to \mathbb{G}$ |
| $H_5$ | $\mathbb{G} \to \mathbb{Z}_q$ |

initiated and when the membership of the group associated with the condition value is changed. In essence, the originator has to multicast the condition value to the members of the user group. However, since this can be done implicitly by the cloud infrastructure, the overhead caused by such multicasting is significantly smaller compared to the direct multicasting done by the originator toward all of the members through a network infrastructure. The originator encrypts the condition value using the public key of a client who is the member of the sharing group for each client. Then, the originator uploads the encrypted condition value to the cloud along with encrypted data. The client who wants to share the data, can download the encrypted data with the encrypted condition value. For the efficient management, a key self-update scheme such as LKH (Logical Key Hierarchy) can be employed.

# 5. ANALYSIS OF PROPOSED SCHEME

## 5.1 Security

The content sharing method proposed is designed on the basis of CPRE. Therefore, the security of proposed method is depending on the security of CPRE. This paragraph analyzes the security of the method proposed on the security model in Section 3.2. The main purpose of the method proposed is to protect data sharing process. Therefore, we need to prove that the proposed scheme does not expose a data during re-encryption process and client without permission cannot obtain a plaintext.

THEOREM 1. *A cloud cannot obtain plaintext by carrying out multiple re-encryptions.*

PROOF. The proposed scheme was designed based on n-Quotient Bilinear Diffie-Hellman Assumption. Therefore, it is computationally secure by complexity of DBDH. □

THEOREM 2. *A client cannot recover the plaintext through chosen ciphertext attack.*

PROOF. The CPRE scheme should meet the following three criteria to have security against selective ciphertext attacks. The fact that level 2 ciphertext is valid should be verified openly. Otherwise, it can be vulnerable to attacks described in [12]. Second, an attacker should not manipulate level 2 ciphertexts maliciously. Third, Level 1 ciphertexts should not be manipulated by an attacker.

The scheme proposed in this study is designed according to Weng et al. and follows the methods applied by them to resist against a selective plain text attack. As security of the scheme against a selective plain text attack has been already proved in Weng et al. we do not repeat verification in this study. In addition, 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 consist of two values, $(rk_1, rk_2)$ and by including $H_5(pk_B^{s \cdot sk_i}) = H_5(pk_2^{s \cdot sk_i})$ 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 [8]. □

THEOREM 3. *A client cannot recover the plaintext without condition values.*

PROOF. The scheme proposed in this study controls decryption authority with condition values. A receiver would not recover data unless he/she knows a condition value even if he/she has re-encryption keys. As a condition value is included in a ciphertext that is issued with CPRE, security of the condition value is the same as that of the ciphertext. As the security of a ciphertext against a selective plaintext attack has been already proved, it can be said that a security of a ciphertext is proved at least in calculation. Next, the case that a user who receives re-encryption keys without a condition value obtains a plain text through a brute force attack should be considered. To acquire security against a brute force attack, a sufficient length of a condition value should be used in realizing the scheme

**Table 2: Overhead Comparison when the condition value changed**

| | CPRE | Proposed Scheme |
|---|---|---|
| Re-encryption key generation | $n\times$(4 exponential operations + 2 multiplications + 1 hash operation) | 1 |
| Encryption of the condition value | 2 exponential operations + 1 multiplication | $n\times$(2 exponential operations + 1 multiplication) |

proposed. □

## 5.2 Efficiency

We analyze the efficiency of our scheme in two different aspects.

First, we discuss about the efficiency of the proposed encryption scheme in terms of secure data sharing in cloud computing circumstance. It is expected that the proposed scheme has better efficiency than other encryption scheme when data is shared through a cloud server. The greatest advantage of our scheme is a user can save cost of expensive re-encryption process by delegating it to sever. In fact, the role of a user under the circumstance that data sharing frequently happens can be limited to encryption and transfer of data to a server. Burden of a user can be reduced even if a user receives heavy data sharing requests because operation necessary to share data is done by a server. This can diminish burden of a client, compared to existing schemes such as ABE or group key based schemes.

Second, we discuss the efficiency of the proposed scheme in terms of the number of re-encryption keys issued after the change of the membership of the group. In previous CPRE schemes, an originator has to generate a re-encryption key (for each user) which is associated with a condition value. Each time the originator wants to revoke a certain member, the re-encryption key has to be re-generated by the originator. In our scheme, we only each ciphertext is associated with a condition value, but each re-encryption key is not. Also, the condition value is distributed to the other users via cloud. This means that the cost to distribute the condition value is minimal. Also, the originator needs to generate the re-encryption key of a user for only one time regardless of the change of the membership of the group.

Compared with previous CPRE schemes, our scheme encrypts the condition value $n$ times when condition value is changed, where $n$ is the number of members in the data sharing group. CPRE generates re-encryption key $n$ times instead of encrypting condition value. To be more precise, re-encryption key generation process of CPRE needs more computational overhead. Therefore, we argue that our scheme is more efficient than previous schemes as shown in the Table 2. The computational overhead about encryption of the condition value was based on ElGamal encryption.

# 6. CONCLUSION

This study proposes a content sharing scheme that is safe in the cloud computing environment, based on a conditional proxy re-encryption scheme. The scheme proposed in this study can significantly reduce burden of a client due to two charateristics. First, re-encryption process is delegated to a cloud server. A client is only involved in process of encryption and decryption of data and creation of re-encryption keys. Second, the number of re-encryption keys to be required for sharing is minimized. In existing CPRE schemes, a condition value is included in a re-encryption key. Therefore, the number of re-encryption keys increases as the number of condition values rises. For the scheme proposed in this study, the increased number of condition values does not result in the increased number of re-encryption keys as re-encryption keys are included in a ciphertext.

# 7. ACKNOWLEDGMENTS

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-0833184 and by US Army Research Office (ARO) No. W911NF-0810510.

# References

[1] Khan, K. M., and Malluhi, Q. 2010 Establishing Trust in Cloud Computing. *IT Professional 12*, 5, 20–27.

[2] Takabi, H., Joshi, J., and Ahn, G. 2010 Security and Privacy Challenges in Cloud Computing Environments. *IEEE Security & Privacy 8*, 6, 24–31.

[3] Kaufman, L. M. 2009 Data Security in the World of Cloud Computing. *IEEE Security & Privacy 7*, 4, 61–64.

[4] Ren, K., Wang, C., and Wang, Q. 2012 Security Chanllenges for the Public Cloud. *IEEE Internet Computing* bibemph16, 1, 59–73.

[5] Ateniese, G., Fu, K., Green, M., and Hohenberger, S. 2006 Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage. *ACM Transactions on Information and System Security (TISSEC) 9*, 1, 1–30.

[6] Mambo, M., and Okamoto, E. 1997 Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertext. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (TFECCS)*, 54–63.

[7] Blaze, M., Beumer, G., and Strauss, M. 1998 Divertible Protocols and Atomic Proxy Cryptography. *Proceeding of International Conference on the Theory and Application of Cryptographic Techniques (Eurocrypt)*, 127–144.

[8] Weng, J., Yang, Y., Tang, Q., Deng, R. H., and Bao, F. 2009 Efficient Conditional Proxy Re-encryption with Chosen-Ciphertext Security. *Proceedings of the 12th Information Security Conference (ISC)*, 151–166.

[9] Chu, C. K., Weng, J., Chow, S. S. M., Zhou J., and Deng, R. H. 2009 Conditional Proxy Broadcast Re-Encryption. *Proceedings of the 14th Australasian Conference on Information Security and Privacy (ACISP)*, 327–342.

[10] Yu, S., Wang, C., Ren, K., and Lou, W. 2010 Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing. *Proceedings of the 29th IEEE International Conference on computer Communications*, 534–542.

[11] Lu, R., Lin, X., Liang, X., and Shen, X. 2010 Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing. *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, 282–292.

[12] Deng, R. H., Weng, J., Liu, S., and Chen, K. 2008 Chosen-Ciphertext Secure Proxy Re-encryption without Pairings. *Proceedings of the 7th International Conference on Cryptology and Network Security (CANS)*, 1–17.

[13] Boneh, D., Lynn, B., and Shacham, H. 2001 Short Signatures from the Weil Pairing. *Proceedings of the 7th International Conference on Theory and Application of Cryptology and Information Security (ASIACRYPT)*, 514–532.

[14] Cash, D., Kiltz, E., and Shoup, V. 2008 The Twin Diffie-Hellman Problem and Applications. *Proceeding of International Conference on the Theory and Application of Cryptographic Techniques (Eurocrypt)*, 127–145.

[15] Zhao, J., Feng, D., and Zhang, Z. 2010 Attribute-Based Conditional Proxy Re-Encryption with Chosen-Ciphertext Security. *Proceeding of Global Telecommunications Conference (GLOBECOM)*, 1–6.