International Journal of Grid and Distributed Computing
Vol. 11, No. 10 (2018), pp.29-40
http://dx.doi.org/10.14257/ijgdc.2018.11.10.03

# Secure Data Sharing using Proxy Re-Encryption for Intelligent Customized Services

Hyun-Jong Cha[1], Ho-Kyung Yang[2] and You-Jin Song[*]

*[1,2]Division of Information Technology Education, Sunmoon University, 70, Sunmoon-ro 221 beon-gil, Tangjeong-myeon, Asan-si, Chungcheongnam-do, 31460, Korea*
*[*]Dept. of Management, Dongguk University, 707, Seokjang-dong, Gyeongju, Gyeongsangbuk-do, 38066, Korea*
*[1]chj826@kw.ac.kr, [2]porori0421@naver.com, *song@dongguk.ac.kr*

***Abstract***

*Intelligent customized services are required to meet the needs of users. Data concerning the user's environment is needed more so the situation of the user can be determined for providing a highly reliable service. In order to collect a large amount of data, the Internet of Things (IoT), which connects numerous devices through the Internet, has been developed. Cloud computing is widely used for managing data efficiently, but can delay data transmission. Therefore, fog computing has been proposed. However, the method used to manage data securely in the cloud computing environment is not efficient in the fog computing environment. Moreover, data may contain personally sensitive information. Additionally, various stakeholders may exist in relation to the access authority of the information. In this respect, there can be many vulnerabilities and threats. In response, this study proposed a re-encryption technique for managing data securely in the fog computing environment. This technique allows a user to delegate the decryption authority to decrypt encrypted data using the re-encryption method in the proxy server for efficient communication.*

***Keywords:*** *Fog Computing, Proxy Re-encryption, IoT, Data Sharing*

## 1. Introduction

Various attempts have been made to apply distributed computing technology to process and treat large amounts of raw data, store and manage structured information, and extract useful information resulting from increased amounts of Internet service data. However, since the traditional network infrastructure is implemented mainly for communication between people, it is difficult to effectively accommodate the variety of types of data traffic in real time. Whereas communication between humans proceeds with a relatively predictable traffic pattern, data traffic in the age of the Internet of Things has a wider variation, which makes it difficult to effectively handle traffic with the current network infrastructure technology.

In the near future, the environment of the Internet of Things (IoT) will come[1-4]. This environment will support communication between an object and a person, and objects without the intervention of people, as well as between people using terminals such as computers and smartphones. In the IoT environment, objects in the physical world are able to sense themselves and the sensed information can be shared via the Internet [5, 6]. Moreover, diverse services can be provided through the interactions between people and objects, and objects and objects [5, 6]. As the IoT develops, IoT devices, IoT service, and

Received (May 12, 2018), Review Result (July 21, 2018), Accepted (September 6, 2018)
* Corresponding Author

ISSN: 2005-4262 IJGDC

IoT users are increasing daily. Cloud computing is widely used along with the IoT in order to store and process the massive data generated by IoT devices. However, the massive data generated by IoT devices in the cloud computing environment has transmission delays because it is stored in a cloud server located far away. Fog computing has emerged to solve this problem. Fog computing can provide better services because it has a lower possibility to cause transmission delays as a cloud server of fog computing is located closer to the user area than that of cloud computing. However, what is critical in the intelligent customized service, which is important in the IoT, is to share data safely and securely.

Therefore, data processing, sharing, and the utilization of network infrastructure should be performed more actively in conjunction with the emergence of the IoT. Furthermore, various services such as real-time connectivity, low response time, and security enhancement will be required to realize the ideal appearance of the IoT in that it can transfer data at any time and any place freely and utilize data for diverse services.

While the existing network infrastructure has been used only as a path for data transmission between a data center and a terminal, the future network infrastructure will manage the whole process from the production to the utilization of data created by various objects. Moreover, it will be able to autonomously conduct various services, even major services supported by data centers. Particularly, it may take a lot of processing time for the security functions, such as encryption, for sharing and utilizing the data stored in data centers securely.

Security services are required for the network edge, in proximity to the terminals, with considering the proximity to the users, high-density geographical distribution, and mobility support, unlike the current cloud computing architecture supporting various application services from central data centers. This study proposed a method to share data securely by re-encrypting at the proxy of the edge to reflect these demands.

Fog computing involves a paradigm that extends cloud computing services to the edge of the network and can provide services where terminal equipment is used directly, such as at network edges, set-top boxes or access points. As described above, this has the advantage of providing intelligent security services that reduce service delay by providing services from a place near the user, which improves quality according to user needs or environment.

As such, the management of distributed data becomes a major issue as various data services become available in the fog computing environment. Security vulnerabilities and privacy violations by malicious attackers or internal users can arise from various types of data transactions (such as sharing and utilization). In the future, as the amount of information handled by e-governments and private enterprises increases, sensing information must be digitized, and information sharing will be essential to enable intelligent services. However, there may be various vulnerabilities and threats as the sensing information can include personal information and there can be a number of stakeholders in information access rights. Moreover, the secondary sharing and utilization of data have the unique trait that many unspecified people, whose relationships are not set in advance, participate. It is expected that many different data services will be provided and sensitive data ownership will be at risk of infringement. Above all, the sharing and utilization of data services will be generalized in the IOT regardless of the user's wishes. In this way, because sensing information is digitized and stored by a centrally managed controller in the data center, if it is improperly used by the user, it can lead to a violation of the individual's privacy. Therefore, in order for intelligent services to be activated, it is necessary to protect the confidentiality of information and manage access rights.

When data is encrypted for sharing data securely and reducing security risks, the conventional encryption method can be a restriction in terms of distribution and management. For example, the use of the AES cryptosystem [8] has issues associated with the secret code management and distribution.

 

When data was encrypted, this study allowed a person who was delegated an authority to decrypt using the re-encrypted secret key instead of distributing the existing secret key to decrypt, which is the conventional method of using the proxy re-encryption method. This delegation of authority will provide the secure sharing and utilization environment while managing data.

This study will review the proxy re-encryption technique that can manage data securely by delegating the decryption authority while managing encrypted data in the fog computing environment. This study is composed as follows: Chapter 2 will review the AES encryption and proxy re-encryption methods as related studies. Chapter 3 will review the ID-based proxy re-encryption method that can delegate decryption authority. Chapter 4 will discuss management of data access authority. Chapter 5 will summarize the conclusions of this study.

## 2. Related Research

### 2.1. Fog Computing

Fog computer is a paradigm that extends cloud computing from the center of the network to the edge of the network. It arranges fog servers between the central cloud server and the terminal smart devices to provide user service as shown in Figure 1. Cloud computing shares computing resources, such as servers and storage, and allows users to access the shared resources through networks. Therefore, it provides services that meet user needs promptly.

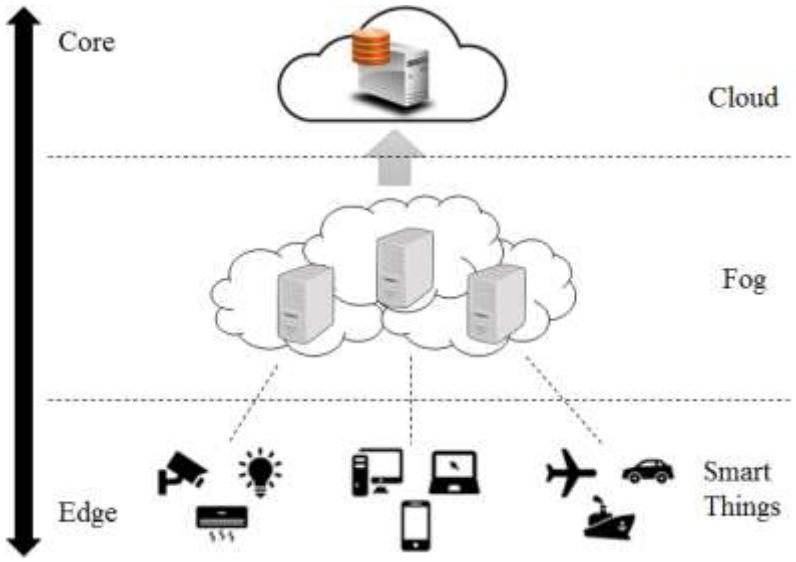


**Figure 1. Fog Computing**

Fog computing has features of cloud computing such as on-demand service, broadband network access, and fast elasticity. At the same time, it has many distinct features [9][10]. The first feature is that it lowers the frequency of delay occurrence because it is located near the edge. This feature is the primary background of the emergence of fog computing. Moreover, it is for smooth supporting services to nodes at the edge of numerous networks that are widely physically distributed.

The second feature is that it can support interactions and mobility through real-time processing. Unlike centralized cloud computing, the terminal devices, the service targets of fog computing, are widely distributed. This environment requires interactions through a real-time process rather than a batch process. Moreover, it requires mobility support because mobile devices account for the majority of terminal devices.

 

The last feature is that it can interoperate in heterogeneous environments. Since fog servers are physically distributed and deployed in different environments, the cooperation between companies is necessary to support a specific service. It should be interoperable in various environments offered by different companies, and should be applicable to different domains.

### 2.2. Advanced Encryption Standard Method

The advanced encryption standard (AES) is a block cipher algorithm that has a symmetric key structure with a "Rijndael" algorithm, which was proposed by Belgian cryptographers V. Rijmen and J. Daemen [8]. The Rijndael encryption algorithm was selected as a standard AES in October 2000. The size of the AES block is a fixed size block of 128 bytes. The size of the secret key can be selected among 128, 192, and 256 bytes. Therefore, it encrypts the 128-byte plaintext block into the 128-byte encryption.

The AES algorithm describes a necessary computation using a $GF(2^3)$ finite field operation and $GF((2^2)^4)$ finite field operation. Hardware implementation is very convenient because the finite field operation does not require carry transmission, unlike the integer operation. The AES algorithm defines the length of the input/output block as 128 bytes and it is displayed as $N_b=4$. This defines the number of 32-bit words. Moreover, the length of the encryption key K is 128, 192, or 256 bytes and is expressed as $N_k$=4, 6, and 8, respectively. Lastly, the number of rounds made during the operation of the algorithm depends on the length of the key. The number of rounds is expressed as $N_r$. When $N_k$ is 4, 6, and 8, $N_r$ is presented as 10, 12, and 14, respectively (Table 1).

**Table 1. Key-block-round Relationship**

| Classification | Key length ($N_k$ words) | Block length ($N_b$ words) | Number of rounds ($N_r$ words) |
|---|---|---|---|
| AES-128 | 4 | 4 | 10 |
| AES-192 | 6 | 4 | 12 |
| AES-256 | 8 | 4 | 14 |

The AES algorithm uses a round function for encryption and decryption operations. The round function is composed of four conversions (*i.e.*, ByteSub, ShiftRow, MixColumn, and AddRoundKey). The following operation is performed for a state composed of 4 rows and 4 columns: the state means the conversion of 128 bytes of data into a two-dimensional array featuring 4 x 4 bytes.

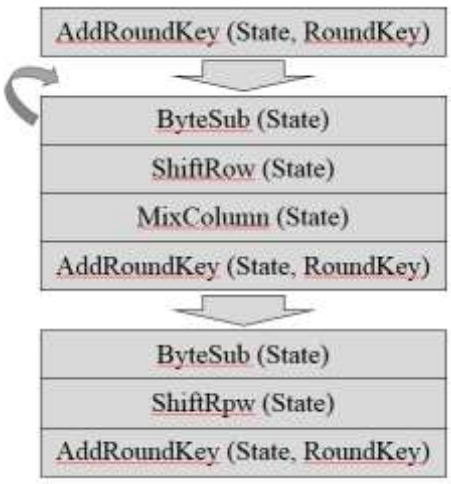


**Figure 2. AES Encryption Course**

The encryption process of AES follows each step in Figure 2. Although the decryption process and the encryption process use the same algorithm, the process order are reversed. Moreover, the transformation process used for the decryption of each detailed process uses a conversion table that is the inverse relationship of the change process used for encryption. It expresses the byte values of the current bytes of the block, which is the input of each process step, and describes the sub-key used in each round as RoundKey.

### 2.3. Identification Based Encryption Method

**2.3.1. Bilinear Mapping:** For two Cyclic Groups $G_1$, $G_2$, bilinear mapping $e: G_1 \times G_2 \rightarrow G_T$ (where $G_T$ is the output space of the bilinear mapping) has the following properties:

- Bilinear property: For all $u \in G_1$, $v \in G_2$ and all $a, b \in Z$, the equation $e(u^a, v^b) = e(u, v)^{ab}$ is valid.

- Non-degenerate property: For $G_x (x = 1, 2)$, generator $g \in G_x$, $e(g, g) \neq 1$.

- Computable property: For all $u \in G_1$, $v \in G_2$, there exists an efficient algorithm to compute $e(u, v)$.

**2.3.2. Complexity Assumptions:** ID-based cryptography has proven to be secure against chosen-plaintext attacks (CPAs) in a random oracle model, under the assumption that it is difficult to compute the Bilinear Diffie–Hellman (BDH) assumption problem. Following is the result of examining the BDH assumption, which is the basis of the ID-based cryptosystem's security.

- Decisional BDH Assumption

Arbitrarily set $g$, $g_a$, $g_b$, $g_c \in G$, $T \in G$. Neither $\{g, g_a, g_b, g_c, e(g,g)_{abc}\}$ nor $\{g, g_a, g_b, g_c, T\}$ can be identified with a probability of 1/2 or more by the algorithm within polynomial time.

- Computational BDH Assumption

Arbitrarily set $g$, $g_a$, $g_b$, $g_c \in G$. Compared to this value, $e(g,g)^{abc}$ cannot be calculated using the algorithm within polynomial time.

ID-based cryptography was first proposed by A. Shamir [1] in 1984. In this scheme, the sender encrypts data using the ID of the recipient, and the ciphertext recipient derives the secret key from PKG and decrypts it.

ID-based cryptography generates a public key from the user's well-known information, such as publicly available information (e mail, IP address, etc.) that identifies the recipient; this is the recipient's ID. Boneh Franklin's ID-based encryption scheme [3] consists of the following four algorithms.

(1) Setup $(k)$: An algorithm for inputting the security parameter $k$ and outputting the public parameter $params$ and the master key $s$ corresponding to the value.

- $[q, G_1, G_2, e] \leftarrow G(k)$, $P \leftarrow G_1$, $s \leftarrow Z_q^*$, $P_{pub} = sP$ is generated

- $H_1: \{0,1\}^* \rightarrow G_1^*$ and $H_2: G_2 \rightarrow \{0,1\}^n$.

- $params = \langle q, G_1, G_2, e, n, P, P_{pub}, H_1, H_2 \rangle$ ; master key is $s$.

(2) KeyGen ($ID$, $params$, $s$ ): An algorithm for inputting the master key $s$, recipient's $ID$, and outputting the secret key $d_{ID}$ corresponding to the $ID$.

 

- $d_{ID} = sQ_{ID}$ is generated, $Q_{ID} = H_1(ID)(\in G_1^*)$.

(3) Enc ($params$,$ID$, $m$ ): An algorithm that inputs the public parameter $params$, the recipient's $ID$, the plaintext $m$, and outputs the ciphertext $c$ corresponding to the plaintext.

- Select $Q_{ID} = H_1(ID)$ and $r \leftarrow Z_q^*$ randomly, and calculate $c = \langle rP, m + H_2(g_{ID}{}^r)$, $g_{ID} = e(Q_{ID}, P_{pub})$.

(4) Dec ($params$, $c = \langle U, V \rangle$, $d_{ID}$ ): by inputting the secret key $d_{ID}$ and ciphertext $c$, the plaintext $m$ is output by the algorithm $m = \mathrm{V} + H_2(e(d_{ID}, U))$ ; this corresponds to the ciphertext.

### 2.4. Proxy Re-encryption Schemes

Proxy re-encryption schemes are being researched extensively [4-8]. Proxy re-encryption schemes convert the ciphertext so that the proxy can use Bob's secret key to decrypt the ciphertext, which has been encrypted with Alice's public key. In other words, Alice creates a re-encryption key which delegates the decryption authority for her cryptogram, and sends it to a proxy server. Then, the proxy server converts the cryptogram of Alice to Bob in the form that can be decrypted using the secret key of Bob using the re-encryption key.

At this point, because the proxy can convert the ciphertext without decrypting the existing ciphertext using the re-encryption key to convert the ciphertext, the proxy knows neither the plaintext nor Alice's secret key. This method can be applied to the transmission of encrypted mail and file systems. For example, in the case of mail transmission, if Alice is absent or the secret key is lost, it is possible for the proxy to convert Alice's encrypted email into Bob's encrypted email and then send. At this time, the proxy does not decrypt the ciphertext but only converts Alice's ciphertext to Bob's ciphertext. In addition, Bob can decrypt the ciphertext with his own secret key without using Alice's secret key (Figure 3).

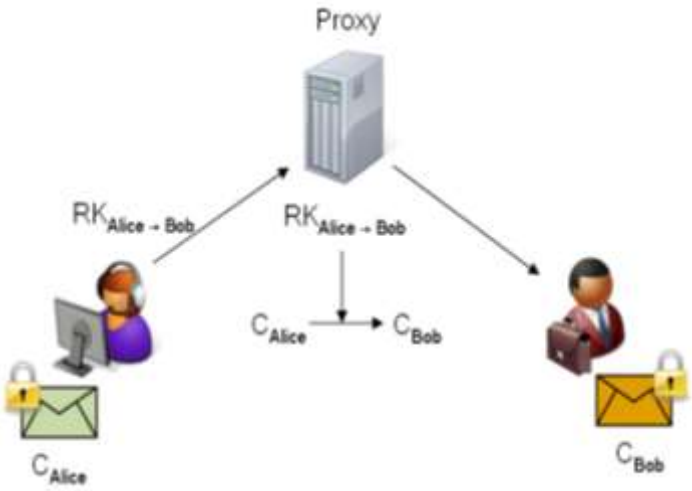


**Figure 3. Example of Mail Transmission based on Proxy Re-Encryption**

## 3. Proposed System

This section reviews the ID-based cipher that has a proxy re-encryption function [8][11]. The ID-based proxy re-encryption method converts the cipher encrypted by the proxy using Alice's ID to the cipher encrypted by Bob's ID.

 

- The proxy converts the plaintext using the re-encryption key without recognizing the context of the text.

- It is impossible to guess any information about Alice's and Bob's secret key from the re-encryption key.

### 3.1. Proxy Re-encryption Method

In 2006, Green and Ateniese [8] proposed the IBE with a Proxy Re-encryption function (GA scheme). The GA scheme has a multi-use property, meaning it can convert Alice's ciphertext to Bob's ciphertext and then convert Bob's ciphertext to Chris's ciphertext.

However, it has a disadvantage in that the length of the ciphertext increases every time the ciphertext is converted.

The GA scheme [8] is based on Boneh and Franklin's IBE scheme, and it is secure because it is based on DBDH in the random oracle model. It consists of the six algorithms of Setup, KeyGen, Encrypt, RKGen, Re-encrypt, and Decrypt:

(1) Setup( $k$ ) : bilinear mapping $e: G_1 \times G_2 \to G_T$ sets $G_1 = \langle g \rangle$ and $G_T$ , and $H_1 : \{0,1\}^* \to G_1$ and $H_2 : G_T \to G_1$ with prime $q$. $s \leftarrow G_T$ is randomly selected.

- $params = (G_1, H_1, H_2, g, g^s)$.

- $msk = s$.

(2) KeyGen( $params$, $msk$, $id$): by inputting identifier $id \in \{0,1\}^*$ and $msk$, the secret key is generated.

- $sk_{id} = H_1(id)^s$.

(3) Encrypt( $params$, $id$, $m$): Encrypts the plaintext $m$ with the identifier $id$, and randomly selects $r \leftarrow Z_q^*$ to encrypt.

- $c_{id} = \left(g^r, m \cdot e\left(g^s, H_1(id)\right)^r\right)$.

(4) RKGen ($params$, $sk_{id_1}$, $id_2$): randomly selects $X \leftarrow G_T$, and by using the identifier $id_2$, encrypts $X$ based on IBE. Then, by using the secret key of $id_1$, generates re-encryption key $rk_{id_1 \to id_2}$.

- $\langle R_1, R_2 \rangle = Encrypt(params, id_2, X)$.

- $rk_{id_1 \to id_2} = \langle R_1, R_2, sk_{id_1}^{-1}, H_2(X) \rangle$.

(5) Re-encrypt ($params$, $rk_{id_1 \to id_2}$, $c_{id_1}$): encrypts $c_{id_1} = c_1, c_2$ with the re-encryption key, and then generates $c_1'$, $c_2'$, $c_3'$.

- $c_1' = c_1$.
- $c_2' = c_2 \cdot e\left(g^r, rk_{id_1 \to id_2}\right)$

$$= m \cdot e\left(g^s, H_1(id)\right)^r \cdot e\left(g^r, H_1(id)^s, H_2(X)\right)$$

$$= m \cdot e\left(g^r, H_2(X)\right).$$

- $c_3' = R_1, R_2$.

(6) Decrypt( $params$, $sk_{id}$ , $c_{id}$): Restores the ciphertext $c_1'$, $c_2'$, $c_3'$ using the secret key. First, generate $X$ by restoring $c_3' = R_1, R_2$ with its own secret key, and then computes the following to get plaintext:

$$- \quad \frac{c_2'}{e\left(c_1', H_2(X)\right)} = \frac{m \cdot e(g^r, H_2(X))}{e(g^r, H_2(X))} = m.$$

The multi-proxy re-encryption scheme divides a portion of the ciphertext encrypted by the delegator's identification information into shares using the multi-proxy identification information. It generates a portion ($U'$) of the ciphertext as a value restored by the share, and re-encrypts the key to be decrypted with the secret key of the delegate via the re-encryption key.

### 3.2. Data Access Management

Let's assume a case involving the use of massive encrypted data. We assume a distributed system that requires a decrypted key for encrypted content from a proxy (key server and content exchange) when using the encrypted content.

In the conventional proxy method, the owner of the encrypted content has to trust that the proxy is safe. Moreover, he or she must trust that it is completely controlled so that the operator of the proxy cannot know any information about the key.

The application of PRE(Proxy Re-Encryption) that modifies the AFGH re-encryption method decreases the dependency of trust required for the proxy. In other words, only the content owner can delegate access to the file without sharing any secret key in the untrusted file server through the AFGH method and the server operator cannot access the stored key (Figure 4).

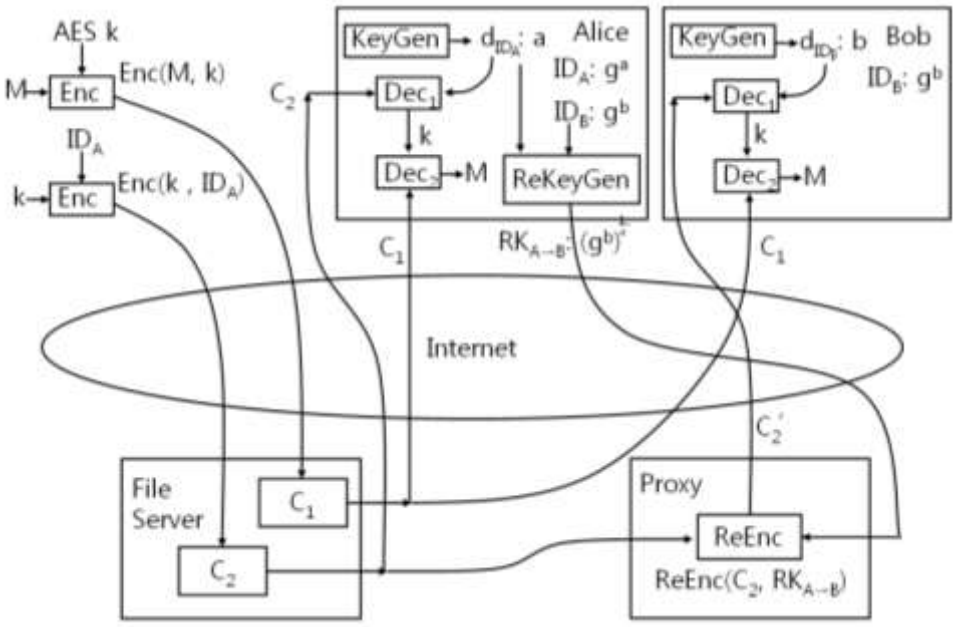


**Figure 4. Outline of Proposed Technique**

The flow of the distributed access control function is shown in Figure 5.

 

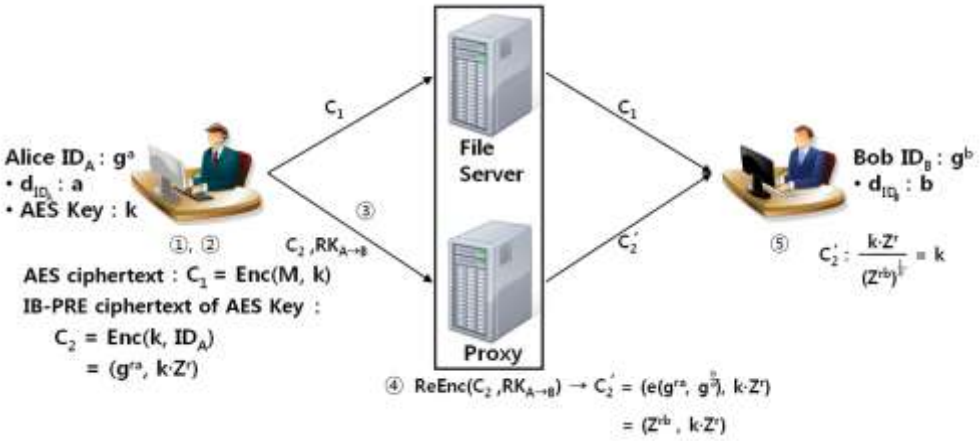


**Figure 5. PRE-based Data Access Management**

(1) The data owner (Alice) generates cryptogram $C_1$ by encrypting massive data ($M$) using AES symmetric key ($k$) in the AES encryption method. At this time, a separate session key $k$ is used for each data $M$.

(2) Alice generates $C_2$ by encrypting the secret key $k$, which is used to encrypt data $M$ using her own open key $ID_a$ (data identifier). Afterward, she sends $C_1$ to the file server of the ITS provider (can be the same host with the proxy) and $C_2$ to the proxy.

(3) When the data user (Bob) requests access to the data $M$, Bob requests the re-encryption of $C_2$ using the re-encryption $RK_{A\to B}$ , which was generated by Alice and sent to the proxy.

(4) The proxy generates $C_2{}'$ by re-encrypting $C_2$ using $RK_{A\to B}$.

(5) Bob receives $C_1$ from the server to decrypt the data $M$ and decrypts $C_2'$, which is re-encrypted from the proxy. $C_1$ is decrypted using the generated secret key $k$ and data $M$ is decrypted.

## 4. Analysis

### 4.1. Security

The study analyzes two types of online attackers. The type 1 attacker is only capable of tapping. The type 2 attacker is an attacker holding a specific encryption key. This study analyzes the unobservability of a single message, the non-connectivity of multiple messages composed by the same user, and the satisfaction of anonymity in two aspects for security in this attacker model.

First, the attacker type 1 cannot identify a user who sent a single message. The attacker type 2 can identify the user but is not a security threat in reality. If it is assumed that a user cannot be identified from the content of a single message, the user who signed and sent the message cannot be identified unless the ID is decrypted. Therefore, non-observability is guaranteed in the case of attacker type 1. An attacker who has acquired keys can identify the sender of the message by decrypting the ID. However, it is not necessary to consider the possibility of revealing the re-encryption key in reality.

Secondly, the attacker type 1 cannot connect messages sent by the same user using different IDs. The attacker type 2 can connect them but they cannot be a security threat in reality. The ID used by a user is updated using the re-encryption technique. Therefore, no-connectivity is provided between them. Consequently, it provides no connectivity between messages signed by different IDs. An attacker who obtains a specific key can

 

identify the user who sent the message by decrypting the ID of each message. However, as mentioned in the first analysis, it is not necessary to consider this threat in reality.

### 4.2. Efficiency

The size of the message excluding the contents of the message is 219 bytes. Even after adding 64 bytes of the control message, it does not exceed 256 bytes. Therefore, the size of the message used in this study is a reasonable size that can be used practically. The computation cost required(Re-encryption cost, certification cost, MAC cost) to generate each message is three exponents(Expression calculation can be optimized by a cost of one exponent), two multiplications, and three hash operations(consider MAC as cost of two hash operations). Additionally, one exponent and two hash computation costs are required costs for verification. Considering the cost, it is an efficient method compared to the identity-based system using the bilinear [12].

## 5. Conclusion

Fog Computing extends cloud computing from central servers to devices at the end of networks. Since cloud providers cannot be fully trusted in terms of data security management, data security threats and privacy invasion factors must be considered. Particularly, it is necessary to ensure data confidentiality.

In this paper concerning proxy re-encryption, the secure sharing of data required in a fog computing environment has been examined. It is possible to manage and share data securely when decentralizing data by delegating the decryption rights for the ciphertext to others using a proxy re-encryption scheme.

In other words, only users who have the authority can decrypt data by determining the data reading authority.

The proposed method can be used in the IoT environment in which sensitive data can be included while various data are generated from many objects.

In order to enhance privacy in the IoT environment, the data owner must have control authority for his or her own information. In order to meet these requirements, more extended technology is needed. In the future, it is our plan to compare the aspects of efficiency through implementation and to study the attribute-based proxy re-encryption scheme, which delegates decryption rights according to attributes related to the type and kind of sensing data.

## Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2016R1D1A1B03931689). This work was also supported by the Dongguk University Research Fund of 2016.

## References

[1] D. Bandyopadhyay and J. Sen, “Internet of Things: Applications and Challenges in Technology and Standardization”, Wireless Personal Communications, vol. 58, no. 1, **(2011)**, pp.49–9.
[2] D. Miorandi, S. Sicari, F. De Pellegrini and I. Chlamtac, “Internet of things: Vision, applications and research challenges”, Ad Hoc Networks, vol. 10, no. 7, **(2012)**, pp. 1497-1516.
[3] L. Coetzee and J. Eksteen, “The internet of things-promise for the future? An introduction”, IST-Africa Conference Proceedings, Gaborone, Botswana, **(2011)** May 11-13, pp. 1-9.
[4] L. Atzori, A. Iera and G. Morabito, “The internet of things: A survey”, Computer Networks, vol. 54, no. 15, **(2010)**, pp. 2787-2805.
[5] J. Gubbia, R. Buyyab, S. Marusica and M. Palaniswamia, “Internet of Things (IoT): A vision, architectural elements, and future irections”, Future Generation Computer Systems, vol. 29, no. 7, **(2013)**, pp. 1645-1660.

 

[6] C. Sarma, Amardeo and J. Girao, "Identities in the future internet of things", Wireless personal communications, vol. 49, no. 3, **(2009)**, pp. 353-363.
[7] A. Shamir, "Identity-Based Cryptosystems and Signature Schemes", Workshop on the theory and application of cryptographic techniques, pp. 341-349, Springer, Berlin, Heidelberg, **(1984)** November 24.
[8] R. Sakai, K. Ohgishi and M. Kasahara., "Cryptosystems based on pairing", Symposium on Cryptography and Information Security, Japan, **(2000)** January.
[9] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog computing and its role in the internet of things", Proceedings of the first edition of the MCC workshop on Mobile cloud computing, ACM, Helsinki, Finland, **(2012)** August 17.
[10] P. Mell and G. Tim, "The NIST definition of cloud computing", National Institute of Standards and Technology, U.S. Department of Commerce, **(2011)**.
[11] S. Brands, "Untraceable off-line cash in wallets with observers", Advances in Cryptology, Crypto 1993, Santa Barbara, California, USA, **(1993)** August 22-26.
[12] C. Zhang, R. Lu, X. Lin, P. Ho and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks", INFOCOM 2008, The 27th Conference on Computer Communications, IEEE (pp. 246-350), Phoenix, AZ, USA, **(2008)** April 13-18.

## Authors

**Hyun-Jong Cha**, he received the B.S. and M.S in Computer science from Kwangwoon University, Seoul, Korea, in 2005 and 2008. He received the M.S. and Ph.D. in Defense Acquisition from Kwangwoon University, Seoul, Korea, in 2011 and 2014. His research interests include defense acquisition, network security and network control architecture.

**Ho-Kyung Yang**, she received the B.S. and M.S in Computer science from Kwangwoon University, Seoul, Korea, in 2005 and 2007. She received the M.S. and Ph.D. in Defense Acquisition from Kwangwoon University, Seoul, Korea, in 2010 and 2013. Her research interests include defense acquisition, network security and network control architecture.

**You-Jin Song**, he received the Ph.D. in Department of Information Security, Tokyo Institute of Technology University at Japan. He was work and researh about various security service and protocol at ETRI(Eletroincs and Telecommunications Research Institute) from 1988 and 1996 in Korea. He is a Professor in department of business and administration, Dongguk university Gyeongju Campus, Korea from 1996 and now. His research interest are Blockchain and IoT security services and Privacy-awareness and its applications.

 