A Note on One Secure Cloud Computing Based Framework for Big Data Information Management of Smart Grid

Lihua Liu[1] and Zhengjun Cao[2]
(Corresponding author: Zhengjun Cao)

Department of Mathematics, Shanghai Maritime University[1]
Haigang Ave 1550, Shanghai 201306, China
Email: liulh@shmtu.edu.cn
Department of Mathematics, Shanghai University[2]
Shangda Road 99, Shanghai 200444, China

(Received July 22, 2020; Revised and Accepted Apr. 14, 2022; First Online July 3, 2022)

Abstract

We show that the scheme [IEEE TCC, 3(2), 233-244, 2015] is not confidential because the proxy can recover all ciphertexts. It is incompatible with the general scenario of cloud computing, which requires that the client's input and output should be kept secret. We also find that the paradigm of proxy re-encryption in the scheme is misused, and each private key is generated, which leads to the loss of confidentiality.

Keywords: Cloud Computing; Identity-based Encryption; Proxy Re-encryption; Smart Grid

1 Introduction

Smart grids have been extensively studied and gradually adopted, because of their pretty efficiency and reliability over the traditional power grids [10]. But the deployment of smart grids is often limited to small regions (e.g., within a city or a small area), due to the difficulties of gathering, storing, and processing the related information [9]. As we see, it is not easy to manage a huge amount of information received from a large number of front-end intelligent devices. Besides, it is generally required that a smart grid should support real-time information processing.

Cloud computing benefits scientific and engineering applications by supporting a paradigm shift from local to network-centric computing and network-centric content, which enables customers with limited computational resources to outsource large-scale computational tasks to the cloud. In 2013, Liu *et al.* [16] discussed the problem of multiowner data sharing for dynamic groups in the cloud. Chen *et al.* [6,25] investigated on achieving secure role-based access control on encrypted data in cloud storage. Nabeel *et al.* [17] designed a scheme with privacy preserving policy based content sharing in public clouds. In 2016, Khaleel *et al.* [14,22] discussed the possibility of using caching search engine for files retrieval system, and using cloud based technique for blog search optimization.

In 2018, Salinas *et al.* [20,21] presented an outsourcing scheme for large-scale sparse linear systems of equations. Ding *et al.* [8] pointed out that in the Salinas *et al.*'s scheme the cloud server can recover a client's input. Recently, Cao and Markowitch [3] argued that in the discussed scenario it was unnecessary for a client to outsource the problem because he can finish the computations locally. Chiou *et al.* [7] pointed out that the mutual authentication scheme was flawed. Hsien *et al.* [4,5,12,15] have presented some surveys on public auditing for secure data storage in cloud computing. Wang *et al.* [13,24] presented a survey for reversible data hiding for VQ-compressed images. Very recently, Pan *et al.* [18,19,23] put forth some batch verification schemes for identifying illegal signatures, smart card-based password authentication schemes, and data collaboration scheme with hierarchical attribute-based encryption in cloud computing.

In 2015, Baek et al. [1] presented a secure cloud computing based framework for big data information management in smart grids. They also presented a concrete instantiation based on a proxy re-encryption scheme. But we find the instantiation is not of confidentiality. In the proposed proxy re-encryption scheme, the proxy can recover all ciphertexts. The scheme can be greatly simplified, even if it could be applied to some particular case.

2 Proxy re-encryption

The paradigm of proxy re-encryption is introduced by Blaze *et al.* [2], in which a proxy is given a re-encryption key that allows him to turn a message encrypted under

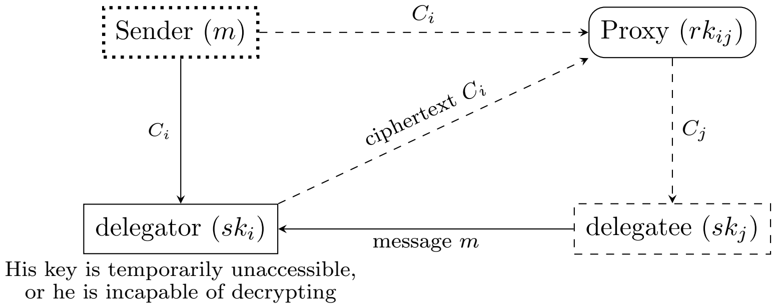

Figure 1: The general proxy re-encryption

a public key pk_1 into a ciphertext of the same message under a public key pk_2, while *the proxy cannot learn anything about the message.* A delegatee only needs to store his own decryption key.

There are many applications of proxy re-encryption, such as forwarding emails and performing operations on storage-limited or power-limited devices, especially, in the scenario that the delegator has less computational power to decrypt, or cannot access to his decryption key temporarily. The delegator can forward a ciphertext to a semi-trusted proxy and ask him to re-encrypt it under a delegatee's public key. The delegatee then decrypts the new ciphertext and returns the plaintext to the delegator (see Figure 1).

Blaze *et al.* [2] also presented a proxy re-encryption scheme which can be described as follows.

Setup. Let $\mathbb{G}$ be a group of prime order p, g be a generator.

Key-generation. The users U_i and U_j choose $x_i, x_j \in \mathbb{Z}_p^*$, and set public keys $X_i = g^{x_i}, X_j = g^{x_j}$ and secret keys x_i, x_j, respectively.

Proxy key generation. U_i computes the proxy key $R_{ij} = x_j/x_i \bmod p$, and sends it to the proxy P.

Encryption. For $m \in \mathbb{G}$ and X_i, the sender picks r to compute the ciphertext $C^{(i)} = (\alpha, \beta) = (X_i^r, g^r \cdot m)$.

Re-encryption. Given $C^{(i)}$, the proxy P computes $\gamma = \alpha^{R_{ij}}$. The re-encryption ciphertext is $C^{(j)} = (\gamma, \beta)$.

Decryption. Given $C^{(i)}$, U_i computes $m = \beta/\alpha^{1/x_i}$. Given $C^{(j)}$, U_j computes $m = \beta/\gamma^{1/x_j}$.

Note that the generation of a proxy key should be elaborated under the collaboration of the delegator and the delegatee, in order to prevent the proxy from recovering plaintexts.

3 Review of Baek *et al.*'s scheme

In 2015, Baek *et al.* [1] presented a scheme for big data information management in smart grids. It involves many entities, the private key generator (PKG), the top cloud (TC), the information storage (IS), Service A, Service B, $\cdots$, and a lot of End-Users (EU). We now describe the scheme (§4.3, [1]) as follows.

Setup. Let $\mathbb{G}_1, \mathbb{G}_2$ be two groups of prime order q, $g \in \mathbb{G}_1$ be a generator, $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ be a bilinear map.

$$\mathcal{H}_1 : \{0,1\}^* \to \mathbb{G}_1, \quad \mathcal{H}_2 : \mathbb{G}_2 \to \{0,1\}^n$$

for some positive integer n, are two hash functions. The PKG picks $s \in \mathbb{Z}_q$ to compute $u = g^s$. Set the master key as s, and the public parameters as $\mathbb{G}_1, \mathbb{G}_2, e, g, u, \mathcal{H}_1, \mathcal{H}_2$.

Key Generation. Given the Top Cloud's identity TC, the Information Storage's identity IS, the Service A's identity $SerA$, and the End-User's identity EU, the PKG computes the private keys

$$\begin{aligned} K_{TC} &= \mathcal{H}_1(TC)^s, \ K_{IS} = \mathcal{H}_1(IS)^s, \\ K_{SerA} &= \mathcal{H}_1(SerA)^s, \ K_{EU} = \mathcal{H}_1(EU)^s, \end{aligned} \quad (1)$$

for the entities, respectively.

Encryption to Top Cloud. See the original description [1].

Encryption to Information Storage. For a message M and the identity IS, an EU picks $r \in \mathbb{Z}_q$ to compute the ciphertext $C_{IS} = (C_1, C_2)$, where

$$C_1 = g^r, \quad C_2 = M \cdot e(u, \mathcal{H}_1(IS))^r.$$

Note that $M = C_2/e(C_1, K_{IS})$.

Proxy Re-encryption Key. For the identity $SerA$, the IS picks $\tilde{r} \in \mathbb{Z}_q$, $T \in \mathbb{G}_2$ to compute the re-encryption key $RK_{IS \to SerA} = (R_1, R_2, R_3)$, where

$$R_1 = g^{\tilde{r}}, R_2 = T \cdot e(u, \mathcal{H}_1(SerA))^{\tilde{r}}, R_3 = K_{IS}^{-1}\mathcal{H}_2(T).$$

Decryption by Service. Given (C_1, C_2), the Service A uses K_{SerA} and (R_1, R_2, R_3) to recover the plaintext

$$M = C_2 \cdot e(C_1, R_3)/e(C_1, \mathcal{H}_2(R_2/e(K_{SerA}, R_1))). \quad (2)$$

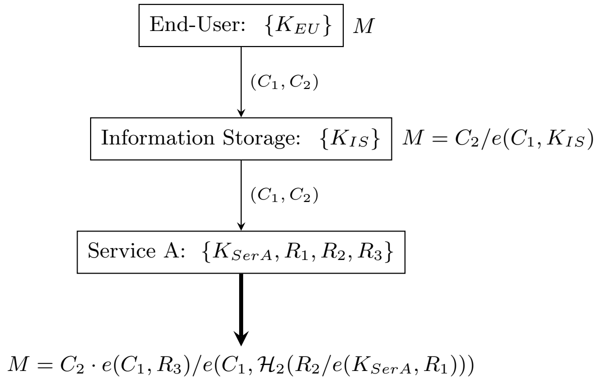
End-User: $\{K_{EU}\}$
M
(C_1, C_2)
Information Storage: $\{K_{IS}\}$
$M = C_2/e(C_1, K_{IS})$
(C_1, C_2)
Service A: $\{K_{SerA}, R_1, R_2, R_3\}$
$M = C_2 \cdot e(C_1, R_3)/e(C_1, \mathcal{H}_2(R_2/e(K_{SerA}, R_1)))$

Figure 2: The data flow in Baek *et al.*'s scheme

4 Analysis

The data flow in the scheme can be depicted by Figure 2.

We now want to stress that the scheme is flawed.

- *The scheme is not of confidentiality.* As we know, in the scenario of cloud computing, it requires that the message M should be protected from the cloud. But in the scheme, except the Information Storage and the Service A, the Top Cloud can also recover the message (§4.3, [1]).

- *The scheme misused the paradigm of proxy re-encryption.* The target receiver, the Information Storage, is also viewed as the proxy (Figure 7, [1]). That means there is no true proxy. In fact, (R_1, R_2, R_3) is solely generated by the IS, and is sent to the Service A. Besides, the ciphertext (C_1, C_2) is directly forwarded to A. Its sketch is just that

$$\left\{\begin{array}{rl} \text{Target receiver} & \xrightarrow[(R_1,R_2,R_3)]{\text{confer}} \text{Other receivers,} \\ \text{Sender} & \xrightarrow[(C_1,C_2)]{\text{encrypt } M} \text{Target receiver} \\ & \xrightarrow[(C_1,C_2)]{\text{forward}} \text{Other receivers.} \end{array}\right.$$

  It is not in accord with the general proxy re-encryption. The phrase "re-encryption" is clearly misused.

- *Each private key in the scheme is simply generated* (see Equation (1)). No sophisticated key-generation skill is adopted. No collaboration for generating keys between participants is necessarily involved. The simple key generation directly results in the loss of confidentiality.

- *The hash function $\mathcal{H}_2 : \mathbb{G}_2 \to \{0,1\}^n$ is falsely specified.* By the definition $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, and the computation $e(C_1, \mathcal{H}_2(R_2/e(K_{SerA}, R_1)))$, we know it should be corrected as $\mathcal{H}_2 : \mathbb{G}_2 \to \mathbb{G}_1$.

- *Even if the scheme could be applied to some extreme case, it can be greatly simplified.* For example, the IS can encrypt M under A's identity as usual. To this end, it suffices to compute the ciphertext $C_{SerA} = (\hat{C}_1, \hat{C}_2)$, where

$$\hat{C}_1 = g^{\hat{r}}, \quad \hat{C}_2 = M \cdot e(u, \mathcal{H}_1(SerA))^{\hat{r}},$$

  $\hat{r} \in \mathbb{Z}_q$ is a random number.

  Given C_{SerA}, the Service A only needs to compute

$$M = \hat{C}_2/e(\hat{C}_1, K_{SerA}). \tag{3}$$

  The new sketch is just that:

$$\begin{array}{rl} \text{Sender} & \xrightarrow[(C_1,C_2)]{\text{encrypt } M} \text{Target receiver} \\ & \xrightarrow[(\hat{C}_1,\hat{C}_2)]{\text{encrypt } M} \text{Other receivers,} \end{array}$$

  which is not of any confidentiality, either. Compared with the original, however, the revision is quite efficient, because the Service A only needs to do one pairing (see Equation (3)), instead of two pairings (see Equation (2)), where $e(K_{SerA}, R_1)$ is independent of any component of a ciphertext, and can be previously computed and stored. By the way, the computation of a pairing is considered somewhat expensive, which has been also discussed to outsource to servers [11].

5 Conclusion

We show that the Baek *et al.*'s scheme is insecure because of its simple key generation. In the proposed scheme, the paradigm of proxy re-encryption is also misunderstood. We want to stress that the confidentiality in the scenario of cloud computing should be carefully considered.

Acknowledgements

We thank the National Natural Science Foundation of China (61411146001). We are grateful to the reviewers for their valuable suggestions.

References

[1] J. Baek and *et al.*, "A secure cloud computing based framework for big data information management of smart grid," *IEEE Transactions on Cloud Computing*, vol. 3, no. 2, pp. 233–244, 2015.

[2] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Proceeding of International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology - EUROCRYPT 1998*, pp. 127–144, Espoo, Finland, June 1998.

[3] Z. J. Cao and O. Markowitch, "Comment on efficient secure outsourcing of large-scale sparse linear systems of equations," *IEEE Transactions on Big Data*, 10.1109/TBDATA.2020.2995200.

[4] W.Y. Chao, C.Y. Tsai, and M.S. Hwang, "An improved key-management scheme for hierarchical access control," *International Journal of Network Security*, vol. 19, no. 4, pp. 639–643, 2017.

[5] J.S. Chen, C.Y. Yang, and M.S. Hwang, "The capacity analysis in the secure cooperative communication system," *International Journal of Network Security*, vol. 19, no. 6, pp. 863–869, 2017.

[6] T.Y. Chen and *et al.*, "Towards secure and efficient user authentication scheme using smart card for multi-server environments," *Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, 2013.

[7] S. F. Chiou and *et al.*, "Cryptanalysis of the mutual authentication and key agreement protocol with smart cards for wireless communications," *International Journal of Network Security*, vol. 21, no. 1, pp. 100–104, 2019.

[8] Q. Ding, G Weng, G. Zhao, and C. Hu, "Efficient and secure outsourcing of large-scale linear system of equations," *IEEE Transactions on Cloud Computing*, 10.1109/TCC.2018.2880181.

[9] Z. Fan and *et al.*, "Smart grid communications: Overview of research challenges, solutions, and standardization activities," *IEEE Communications Surveys Tutorials*, pp. 1–18, 2012.

[10] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010.

[11] M. Girault and D. Lefranc, "Server-aided verification, theory and practice," in *Proceedings of 11th International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology - ASIACRYPT 2005*, pp. 605–623, Chennai, India, December 2005.

[12] W.F. Hsien, C.C. Yang, and M.S. Hwang, "A survey of public auditing for secure data storage in cloud computing," *International Journal of Network Security*, vol. 18, no. 1, pp. 133–142, 2016.

[13] L. C. Huang, T. Y. Chang, and M. S. Hwang, "A conference key scheme based on the diffie-hellman key exchange," *International Journal of Network Security*, vol. 20, no. 6, pp. 1221–1226, 2018.

[14] M. Khaleel, H. El-Bakry, and A. Saleh, "A new efficient files retrieval system using caching search engine," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 22–31, 2016.

[15] C.W. Liu and *et al.*, "A survey of attribute-based access control with user revocation in cloud data storage," *International Journal of Network Security*, vol. 18, no. 5, pp. 900–916, 2016.

[16] X. Liu and *et al.*, "Mona: Secure multiowner data sharing for dynamic groups in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1182–1191, 2013.

[17] M. Nabeel, N. Shang, and E. Bertino, "Privacy preserving policy based content sharing in public clouds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2602–2614, 2013.

[18] H. Pan and *et al.*, "Research on batch verification schemes for identifying illegal signatures," *International Journal of Network Security*, vol. 21, no. 6, pp. 1062–1070, 2019.

[19] H.T. Pan, H.W. Yang, and M.S. Hwang, "An enhanced secure smart card-based password authentication scheme," *International Journal of Network Security*, vol. 22, no. 2, pp. 358–363, 2020.

[20] S. Salinas, C. Luo, X. Chen, and P. Li, "Efficient secure outsourcing of large-scale linear systems of equations," in *Proc. IEEE Conf. Comput. Commun.*, pp. 1035–1043, Apr. 2015.

[21] S. Salinas, C. Luo, X. Chen, W. Liao, and P. Li, "Efficient secure outsourcing of large-scale sparse linear systems of equations," *IEEE Transactions on Big Data*, vol. 4, no. 1, pp. 26–39, 2018.

[22] J. Singh, "Cloud based technique for blog search optimization," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 32–39, 2016.

[23] W.L. Tai, Y.F. Chang, and W.H. Huang, "Security analyses of a data collaboration scheme with hierarchical attribute-based encryption in cloud computing," *International Journal of Network Security*, vol. 22, no. 2, pp. 212–217, 2020.

[24] Y. L. Wang, J. J. Shen, and M. S. Hwang, "A survey of reversible data hiding for vq-compressed images," *International Journal of Network Security*, vol. 20, no. 1, pp. 1–8, 2018.

[25] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure role-based access control on encrypted data in cloud storage," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 1947–1960, 2013.

Lihua Liu, associate professor, with Department of Mathematics at Shanghai Maritime University, received her PhD degree in applied mathematics from Shanghai Jiao Tong University. Her research interests include combinatorics and cryptography.

Zhengjun Cao, associate professor, with the Department of Mathematics, Shanghai University, received his

PhD degree in applied mathematics from Academy of Mathematics and Systems Science, Chinese Academy of Sciences. He had served as a post-doctor in Computer Sciences Department, Université Libre de Bruxelles. His research interests include cryptography, discrete logarithms and quantum computation.