
Wang, X.A. [et al.] (2016) Non-transferable unidirectional proxy re-encryption scheme for secure social cloud storage sharing. 2016 International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2016, 7-9 September 2016, Ostrava, Czech Republic: proceedings. [S.l.]: IEEE, 2016. Pp. 328-331. Doi: http://dx.doi.org/10.1109/INCoS.2016.82.

Wang, X.A. [et al.] (2016) Non-transferable unidirectional proxy re-encryption scheme for secure social cloud storage sharing. 2016 International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2016, 7-9 September 2016, Ostrava, Czech Republic: proceedings. [S.l.]: IEEE, 2016. Pp. 328-331. Doi: http://dx.doi.org/10.1109/INCoS.2016.82.

Non-transferable Unidirectional Proxy Re-encryption Scheme for Secure Social Cloud Storage Sharing

Xu An Wang[1,2], Fatos Xhafa[3], Wei Hao[1], Wei He[4]
[1]Engineering University of Chinese Armed Police Force, P. R. China
[2]School of Cyber Engineering, Xidian University, P. R. China
[3]Department of Computer Science, Technical University of Catalonia, Spain
[4]E-knowledge station of Chinese Armed Police Force, P. R. China
wangxazjd@163.com, fatos@cs.upc.edu

***Abstract*— Proxy re-encryption (PRE), introduced by Blaze et al. in 1998, allows a semi-trusted proxy with the re-encryption key to translate a ciphertext under the *delegator* into another ciphertext, which can be decrypted by the *delegatee*. In this process, the proxy is required to know nothing about the plaintext. Many PRE schemes have been proposed so far, however until now almost all the unidirectional PRE schemes suffer from the transferable property. That is, if the proxy and a set of delegatees collude, they can re-delegate the delegator's decryption rights to the other ones, while the delegator has no agreement on this. Thus designing non-transferable unidirectional PRE scheme is an important open research problem in the field. In this paper, we tackle this open problem by using the composite order bilinear pairing. Concretely, we design a non-transferable unidirectional PRE scheme based on Hohenberger *et al.*'s unidirectional PRE scheme. Furthermore, we discuss our scheme's application to secure cloud storage, especially for sharing private multimedia content for social cloud storage users.**

I. Introduction

Proxy re-encryption scheme was first proposed by Blaze, Bleumer and Strauss [1] in 1998, which allows the proxy to transform a ciphertext for Alice into a ciphertext of the same message for Bob. During the transformation, the proxy learns nothing about the underlying message. In 2005, Ateniese *et a*l. [2] first proposed a unidirectional proxy re-encryption and defined nine interesting notions for proxy re-encryption: unidirectional, non-interactive, proxy invisible, original-access, key optimal, collusion-safe, temporary, non-transitive, non-transferable. Until now, there have not been defined yet PRE schemes satisfying these nine properties simultaneously. Among these properties, non-transferability seems to be a very difficult one to achieve. A proxy re-encryption scheme is said to be non-transferable if the proxy and a set of colluding delegatees cannot re-delegate decryption rights to other parties without compromising any malicious delegatee's decryption capability, in the sense that the only way for Bob to transfer Alice's decryption capability is to expose his own secret key. In all known PRE constructions, if Bob and a malicious proxy collude, they can derive new re-encryption keys for Charlie without Alice's agreement. Non-transferable property can be seen as a trade-off solution to protect Alice's benefit in delegating her decryption right: when a malicious delegatee Bob colludes with the proxy to transfer Alice's decryption capability to others, he must expose his own decryption capability as a pay. This notion emphasis that Bob cannot collude with the proxy and transfer Alice's decryption right without compromising his own decryption capability. Since its introduction, PRE scheme has found many applications, such as key distribution [1], key escrow [3], distributed file systems [2], [4], multicast [5], anonymous communication [6]. Recently, along with the rapid development of cloud computation [7]–[9]. More recently, PRE has been used extensively in secure cloud storage for content sharing such as DFA-based FPRE system [10]–[13].

A. Our Contribution

To the best of our knowledge, until now there are no natural non-transferable proxy re-encryption schemes without obfuscation or involving with PKG. In this paper, we tackle the non-transferable problem by using the composite order bilinear pairing. Concretely, we design a non-transferable scheme based on Hohenberger *et al.*'s unidirectional PRE scheme (Shacham's Linear Encryption Scheme). We also present a secure social cloud storage sharing framework based on our non-transferable proxy re-encryption scheme.

B. Organization

Our paper is organized as the following: In Section II, we first review Hohenberger *et al.*'s PRE scheme and then propose our construction based on it. We also roughly sketch the analysis of our scheme and show it can achieve non-transferability. In Section III, we present a framework for secure social cloud storage sharing based on our proposal. We conclude our paper in the last Section IV.

II. Our Construction

A. Review of Hohenberger et al.'s unidirectional PRE scheme

In this subsection, we review of Hohenberger *et al.*'s unidirectional PRE scheme proposed in TCC'07 [14], based on which we propose our construction.

1) Setup. The scheme operating over two groups G_1, G_2 of prime order q with a bilinear map $e : G_1 \times G_1 \to G_2$. The system parameters are random generators $g \in G_1$ and $Z = e(g, g) \in G_2$.
2) KeyGen. A user A's key pair is of the form $pk_A = (g^{a_1}, g^{b_1}), sk_A = (a_1, b_1)$, a user B's key pair is of the form $pk_B = (g^{a_2}, g^{b_2}), sk_B = (a_2, b_2)$, a user C's key pair is of the form $pk_C = (g^{a_3}, g^{b_3}), sk_C = (a_3, b_3)$, etc.
3) ReKeyGen. A user A delegates to B by publishing the re-encryption key
$$rk_{A\to B} = (Z_1, Z_2) = (g^{a_2/a_1}, g^{b_2/b_1})$$
computed from A's secret key and B's public key.
4) Second-Level Encryption. To encrypt a message $m \in G$ under pk_A in such a way that it can be decrypted by A and her delegatees, the encrypter computes
$$C_A^2 = (W, X, Y) = (g^{a_1 r}, g^{b_1 s}, mg^{r+s})$$
as the ciphertext.
5) First-Level Encryption. To encrypt a message $m \in G_2$ under pk_A in such a way that it can only be decrypted by the holder of sk_A, output
$$\begin{aligned} C_A^1 &= (E, F, G) \\ &= (e(g, g)^{a_1 r}, e(g, g)^{b_1 s}, e(g, m)e(g, g)^{r+s}) \end{aligned}$$
6) Re-Encryption. The proxy can re-encrypt a second-level ciphertext for A into a first-level ciphertext for B with $rk_{A\to B}$. From $C_A^2 = (W, X, Y) = (g^{a_1 r}, g^{b_1 s}, mg^{r+s})$, compute
$$\begin{aligned} C_B^1 &= (E, F, G) \\ &= (e(W, Z_1), e(X, Z_2), e(Y, g)) \\ &= (e(g, g)^{a_2 r}, e(g, g)^{b_2 s}, e(g, m)e(g, g)^{r+s}) \end{aligned}$$
7) Second-Level Decryption. To decrypt a second-level ciphertext $C_A^2 = (W, X, Y) = (g^{a_1 r}, g^{b_1 s}, mg^{r+s})$ with secret key $sk_A = (a_1, b_1)$, A computes
$$m = \frac{Y}{W^{1/a_1} X^{1/b_1}}$$
8) First-Level Decryption. To decrypt a first-level ciphertext $C_A^1 = (E, F, G)$ with secret key $sk_A = (a_1, b_1)$, A first computes
$$Q = \frac{G}{E^{1/a_1} F^{1/b_1}}$$
and then output message m in the message space M such that $e(m, g) = Q$. Of course, to ensure efficient decryption, this limits the size of the message space M to be a polynomial.

B. Our Proposal

1) Setup. The scheme operates over two groups G_1, G_2 of composite order $n = pqh$ with a bilinear map $e : G_1 \times G_1 \to G_2$. The system parameters are random generators $g_p \in G_p$, $g_q \in G_q$, $g_h \in G_H$. that is, g_p, g_q, g_h are generators of the subgroup G_p, G_q and G_h. Furthermore, $e(g_p, g_q) = 1, e(g_p, g_h) = 1, e(g_h, g_q) = 1$ holds.
2) KeyGen. A user A's key pair is of the form $pk_A = (g_p^{a_1}, g_q^{b_1}, g_h^{c_1}), sk_A = (a_1, b_1, c_1)$, a user B's key pair is of the form $pk_B = (g_p^{a_2}, g_q^{b_2}, g_h^{c_2}), sk_B = (a_2, b_2, c_2)$, a user C's key pair is of the form $pk_C = (g_p^{a_3}, g_q^{b_3}, g_q^{c_3}), sk_C = (a_3, b_3, c_3)$, etc.
3) ReKeyGen. A user A delegates to B by publishing the re-encryption key
$$\begin{aligned} rk_{A\to B} &= (Z_1, Z_2, Z_3) \\ &= (g_p^{(a_2+r')/a_1}, g_q^{(b_2+s')/b_1}, g_h^{c_2/c_1} g_p^{r'} g_q^{s'}) \end{aligned}$$
computed from A's secret key and B's public key.
4) Second-Level Encryption. To encrypt a message m under pk_A in such a way that it can be decrypted by A and her delegatees, the encrypter computes:
$$\begin{aligned} C_A^2 &= (V, W, X, Y) \\ &= (e(g_h, g_h)^t, g_p^{a_1 r}, g_q^{b_1 s}, g_h^{mtc_1} g_p^r g_q^s) \end{aligned}$$
as the ciphertext.
5) First-Level Encryption. To encrypt a message m under pk_A in such a way that it can only be decrypted by the holder of sk_A, output:
$$\begin{aligned} C_A^1 &= (C_A^{1a}, C_A^{1b}) \\ &= (e(g_h, g_h)^t, e(g_h, g_h)^{mtc_1}) \end{aligned}$$

6) Re-Encryption. The proxy can re-encrypt a second-level ciphertext for A into a first-level ciphertext for B with $rk_{A\to B}$. From $C_A^2 = (V, W, X, Y) = (e(g_h, g_h)^t, g_p^{a_1 r}, g_q^{b_1 s}, g_h^{mtc_1} g_p^r g_q^s)$, compute:

$$(E, F, G) = (\frac{e(W, Z_1)}{e(g_p^{a_2}, Y)}, \frac{e(X, Z_2)}{e(g_q^{b_2}, Y)}, e(Y, Z_3))$$
$$= (\frac{e(g_p, g_p)^{a_2 r + rr'}}{e(g_p, g_p)^{a_2 r}}, \frac{e(g_q, g_q)^{b_2 s + ss'}}{e(g_p, g_p)^{b_2 s}},$$
$$e(g_h, g_h)^{mtc_2} e(g_p, g_p)^{rr'} e(g_q, g_q)^{ss'})$$
$$= (e(g_p, g_p)^{rr'}, e(g_q, g_q)^{ss'},$$
$$e(g_h, g_h)^{mtc_2} e(g_p, g_p)^{rr'} e(g_q, g_q)^{ss'}))$$

Finally the proxy computes

$$C_B^1 = (C_B^{1a}, C_B^{1b}) = (e(g_h, g_h)^t, e(g_h, g_h)^{mtc_2})$$

7) Second-Level Decryption. To decrypt a second-level ciphertext $C_A^2 = (V, W, X, Y) = (e(g_h, g_h)^t, g_p^{a_1 r}, g_q^{b_1 s}, g_h^{tc_1 m} g_p^r g_q^s)$ with secret key $sk_A = (a_1, b_1, c_1)$, A computes

$$A = g_h^{tc_1 m} = \frac{Y}{W^{1/a_1} X^{1/b_1}},$$
$$A' = e(A, g_h) = e(g_h, g_h)^{tc_1 m}$$

and then output message m in the message space M such that $V^m = (e(g_h, g_h)^t)^m = A'^{1/c_1}$. Of course, to ensure efficient decryption, this limits the size of the message space M to be a polynomial.

8) First-Level Decryption. To decrypt a first-level ciphertext $C_A^1 = (C_A^{1a}, C_A^{1b}) = (e(g_h, g_h)^t, e(g_h, g_h)^{mtc_1})$ with secret key $sk_A = (a_1, b_1, c_1)$, A first computes

$$B = (e(g_h, g_h)^{tc_1 m})^{1/c_1} = e(g_h, g_h)^{tm}$$

and then output message m in the message space M such that $(e(g_h, g_h)^t)^m = B$. Of course, to ensure efficient decryption, this limits the size of the message space M to be a polynomial size.

Remark 1: Why our scheme can achieve non-transferable property? We can see from:

$$rk_{A\to B} = (Z_1, Z_2, Z_3)$$

$$= (g_p^{(a_2 + r')/a_1}, g_q^{(b_2 + s')/b_1}, g_h^{c_2/c_1} g_p^{r'} g_q^{s'})$$

and the delegatee's secret key, other user C's secret key:

$$sk_B = (a_2, b_2, c_2), sk_C = (a_3, b_3, c_3)$$

Thus, the colluders can not compute:

$$rk_{A\to C} = (Z_1', Z_2', Z_3')$$
$$= (g_p^{(a_3 + r'')/a_1}, g_q^{(b_3 + s'')/b_1}, g_h^{c_3/c_1} g_p^{r''} g_q^{s''})$$

III. NT-PRE for Secure Social Cloud Storage Sharing

The secure social cloud storage sharing framework can be designed, using our scheme, as following:

1) Data outsourcing. When Alice travels outside and wants to share her personal multimedia content, e.g. pictures, with others, she first encrypts her own traveling photos by the standard encryption method for JPEG pictures, such as the block cipher, and then encrypts the block cipher key with her own public key. Finally, she outsources the ciphertexts to the cloud, which not only saves her storage but also is convenient for sharing data contents with other persons.
2) Delegation of Re-encryption. When one of her close friends, say Bob, wants to access her photos, he cannot decrypt Alice's ciphertext directly, because the photos are encrypted under Alice's public key. In this case, proxy re-encryption can be used. Alice generates a re-encryption key RK from Alice to Bob and sends it to the proxy, which can be a designated server lying in the cloud.
3) Re-encryption. The proxy first retrieves Alice's ciphertexts from the cloud storage servers, and applies the re-encryption key RK to re-encrypt Alice's ciphertexts into Bob's ciphertexts which Bob can decrypt with his own private key. Then, the proxy sends the re-encrypted ciphertexts to Bob.
4) Data Sharing. Bob uses his own private key to decrypt the re-encrypted ciphertexts, through which Alice shares her photos with Bob in the cloud. We note that during the whole re-encryption process, the sender Alice is online only when sending the re-encryption key. Besides, we also note that our NT-PRE scheme prevents a malicious cloud colluding with Bob to further re-delegate Alice's decryption rights to another user Carol, which guaranteed by the non-transferability property of our scheme (see Fig. 1).

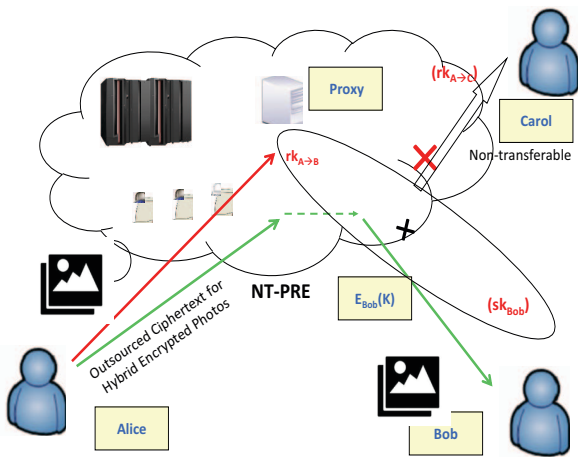

Fig. 1. Non-transferable proxy re-encryption for secure social cloud storage sharing

IV. Conclusion

In this paper we present a new proposal on non-transferable proxy re-encryption. To the best of our knowledge, this proposal is the first natural proxy re-encryption scheme with non-transferable property without obfuscation or involving PKG in the identity based setting. Furthermore, we discuss our scheme's application to social cloud storage sharing, such as secure photos sharing, etc. Our scheme can be used to solve the difficult transferability problem, which can reduce the data owner's worries about their uncontrolled data sharing. There are several interesting aspects that can be further explored, such as proving our proposals' security formally, proposing more efficient non-transferable proxy re-encryption scheme in the prime order group, etc.

V. Acknowledgements

This work is supported by Natural Science Foundation of Shaanxi Province (Grant No. 2014JM8300).

References

[1] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 127–144, Espoo, Finland, May 31 – June 4, 1998. Springer, Berlin, Germany.

[2] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *NDSS 2005*, San Diego, California, USA, February 3–4, 2005. The Internet Society.

[3] Anca Ivan and Yevgeniy Dodis. Proxy cryptography revisited. In *NDSS 2003*, San Diego, California, USA, February 5–7, 2003. The Internet Society.

[4] 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, vol. 9, no. 1, pages 1–30. 2006.

[5] Yun-Peng Chiu, Chin-Laung Lei, and Chun-Ying Huang. Secure multicast using proxy encryption. In Sihan Qing, Wenbo Mao, Javier López, and Guilin Wang, editors, *ICICS 05*, volume 3783 of *LNCS*, pages 280–290, Beijing, China, December 10–13, 2005. Springer, Berlin, Germany.

[6] J. Shao, P. Liu, G. Wei, and Y. Ling. Anonymous proxy re-encryption. Security and Communication Networks, vol. 5, no. 5, pp. 439-449, 2012.

[7] Ying Wang, Jiali Du, Xiaochun Cheng, Zheli Liu, and Kai Lin. Degradation and encryption for outsourced png images in cloud storage. International Journal of Grid and Utility Computing, vol. 7, no. 1, pp. 22-28, 2016.

[8] Ronald Petrlic, Stephan Sekula, and Christoph Sorge. A privacy-friendly architecture for future cloud computing. International Journal of Grid and Utility Computing, Vol. 4, No. 4, pp.265-277, 2013.

[9] Xinfeng Ye and Bakh Khoussainov. Fine-grained access control for cloud computing. International Journal of Grid and Utility Computing, Vol. 4, No. 2/3, pp. 160-168, 2013.

[10] K. Liang, M. H. Au, J. K. Liu, X. Qi, W. Susilo, X. P. Tran, D. S.Wong, and G. Yang. A dfa-based functional proxy re-encryption scheme for secure public cloud data sharing. IEEE Transactions on Information Forensics and Security, vol. 9, no. 10, pp. 1667-1680, 2014.

[11] Kaitai Liang, Joseph K. Liu, Duncan S. Wong, and Willy Susilo. An efficient cloud-based revocable identity-based proxy re-encryption scheme for public clouds data sharing. In Miroslaw Kutylowski and Jaideep Vaidya, editors, *ESORICS 2014, Part I*, volume 8712 of *LNCS*, pages 257–272, Wroclaw, Poland, September 7–11, 2014. Springer, Berlin, Germany.

[12] Matija Puzar and Thomas Plagemann. Data sharing in mobile ad-hoc networks-a study of replication and performance in the midas data space. International Journal of Space-Based and Situated Computing, Vol. 1, No. 2/3, pp. 137-150, 2015.

[13] Meriem Thabet, Mahmoud Boufaida, and Fabrice Kordon. An approach for developing an interoperability mechanism between cloud providers. International Journal of Space-Based and Situated Computing, Vol. 4, No. 2, pp. 88-99, 2014.

[14] Susan Hohenberger, Guy N. Rothblum, abhi shelat, and Vinod Vaikuntanathan. Securely obfuscating re-encryption. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 233–252, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Berlin, Germany.