Cloud Data Sharing using IB-BPRE Based Revocable Scheme

[1]*K. Venkateswarlu* [2] *B. Padmaja*

[1] *Assistant Professor, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.*

[2] *PG Scholar, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.*

Abstract—Cloud computing has become prevalent due to its nature of massive storage and vast computing capabilities. Ensuring a secure data sharing is critical to cloud applications. Recently, a number of identity-based broadcast proxy re-encryption (IB-BPRE) schemes have been proposed to resolve the problem. However, the IB-BPRE requires a cloud user (Alice) who wants to share data with a bunch of other users (e.g. colleagues) to participate the group shared key renewal process because Alice's private key is a prerequisite for shared key generation. This, however, does not leverage the benefit of cloud computing and causes the inconvenience for cloud users. Therefore, a novel security notion named revocable identity-based broadcast proxy re-encryption (RIB-BPRE) is presented to address the issue of key revocation in this work. In a RIB-BPRE scheme, a proxy can revoke a set of delegates, designated by the delegator, from the re-encryption key. The performance evaluation reveals that the proposed scheme is efficient and practical.

Keywords: Proxy Re-Encryption, Cloud Data Sharing, Broadcast Encryption, Revocation.

1. INTRODUCTION

Cloud computing has become a solution for data maintenance due to its flexibility and effectiveness. However, cloud computing has been suffering from security and privacy challenges. Encryption can be a straightforward approach to ensure data confidentiality and Identity-based encryption (IBE) is one of the promising representative secure mechanisms because it has a concise public key infrastructure [1]–[3]. When storing the identity-based encrypted data to the cloud, the data owner would like to share the data with others in particular scenarios.

For example, a set of volunteers upload their genome data to the cloud in a genome record cloud system for the scientists to collaboratively conduct medical research [4]. If IBE is adopted into such a medical system, the genome data should be encrypted before uploading to the cloud as Enc (m, id), where m is the genome data and id is the recipient's identity. A researcher Alice with the identity id from the genome research institute may want to share the volunteer's genome data with a list of her colleagues with identities $id_1,\cdots,id_n$ in the same research group.

Identity-Based Proxy Re-Encryption (IB-PRE)

Proxy re-encryption was proposed to enable a semi-trust proxy to convert a ciphertext with one's identity to a new ciphertext under a different identity [5]. Later on, the notion of IB-PRE [6] was introduced to simplify PKI (Public Key Infrastructure) since the user's identity can be considered as a replacement of the public key in an IB-PRE scheme. One might think the IB-PRE can be a trivial solution to partially address the IBE drawback described in above application in a cloud environment.

For example, Alice, with identity id, can generate a re-encryption key $rk_{id\rightarrow idi,\cdots,}rk_{id\rightarrow idn}$ for each colleague in the delegation list S = $\{id_1,id_2,\cdots,id_n\}$ then she forwards these re-encryption keys to the cloud server. As soon as the server receives the keys, it has the flexibility to re-encrypt the ciphertext for each delegatee accordingly. Moreover, with IBPRE, it is convenient to revoke the individual's re-encryption by simply removing the user from the delegation/revocation list. However, similar to IBE, this solution is very inefficient as Alice is required to compute a re-encryption key for every delegate, in which the number of re-encryption keys is linear to the total counts of delegatees (O(n)). Consequently, IBPRE will not scale well if a huge number of delegatees exist in the group.

Identity-Based Broadcast Proxy Re-Encryption (IBBPRE)

The notion of broadcast proxy re-encryption (BPRE) [7] has been proposed to eliminate the linear computation for re-encryption key generation. Doing so can also resolve the heavy computation issue of IBE. Instead of generating re-encryption key for every single delegatee in the group, a proxy (e.g. a cloud server) only needs to have a broadcast re-encryption key in a BPRE scheme to transform a delegator's ciphertext to a set of delegatees' ciphertext without revealing plaintext to the proxy. Since then, some researchers introduced the notion of identity-based broadcast proxy re-encryption where the user's identity is used as its public key [8]. Despite the potential heavy communication of re-encryption key is resolved by IB-BPRE, key revocation problem still exists in IB-BPRE. Some may argue that Alice can generate a new broadcast re-encryption key as soon as each revocation occurs. As we pointed out earlier, this brings inconvenience to the user Alice since she has to show and present her private key to produce the broadcast re-encryption key. Such a process violates the original intention of cloud computing which is leaving the heavy computing task to the cloud not the user. Moreover, if re-encryption key is leaked in existing IB-BPRE schemes, anybody who obtained the key can re-encrypt the ciphertext. Hence, Alice needs to establish a secure channel to transmit the re-encryption key for each re-encryption key update.

2. RELATED WORK

The primitive of broadcast encryption was first pointed out by Berkovits [9] to enable a sender to broadcast a ciphertext to a set of users and each user from the recipient list is able to decrypt the ciphertext. Fiat and Naor [10] formalized the definition and security model for broadcast encryption. After that, many broadcast encryption schemes were proposed to improve the efficiency [11]–[13]. Sakai and Furukawa [14] presented the notion of identity-based broadcast encryption (IBBE), in which an user's identity is considered as the public key in an identity based broadcast encryption. Delerablee [15] proposed an IBBE scheme with the ciphertext that has a constant size. While IBE offers the convenience on key management, it suffices a limitation of revoking user's identity. Boneh and Franklin [1] gave a seminal solution. In their scheme, the user's public key is replaced by an actual identity id and a separate time period T. Boldyreva, Goyal and Kumar reduced the revocation cost from linear to logarithmic. Recently, Susilo et al. presented an IBBE with a new idea for revocation that supports to directly revoke recipients from the original recipient list. Further, many attribute-based encryption (ABE) were proposed to enable the expression of identity [3].

A notion of proxy re-encryption was proposed to delegate the decryption correctly [5]. Many schemes were proposed to deal with the functionality, efficiency, and security model. Green and Ateniese [6] applied identity-based encryption to proxy re-encryption in an identity-based proxy re-encryption scheme. Subsequently, lots of IB-PRE schemes were proposed mainly to focus on the functionality, efficiency and security. Another interesting research thread is BRPE. For instance, Chu et al. [7] proposed a broadcast proxy re-encryption scheme that enables a proxy to transform Alice's ciphertext to a set of delegates. Following their work, Xu et al. [8] and Sun et al. proposed IB-BPRE schemes in which both their private key and ciphertext have a constant size. Unfortunately, none of these works addressed the re-encryption key revocation issue.

Our Contribution

We adopted the revocation mechanism (recipient revocable) proposed for IBBE to address key revocation issue for IB-BPRE. Although the approach sounds straightforward, there are technical difficulties to apply recipient revocation notion to IB-BPRE because we found the method is vulnerable to the collusion attack. A recipient colludes with the proxy can reveal the delegator's private key. Details can be found in Appendix A. Other than this recipient revocable method, one possible attempt is the approach proposed in [8]. In their scheme, another more N elements should be added in the public key for randomness. When generating a re-encryption key, a user Alice introduced a polynomial for variable µ with degree less than N to randomize her private key. Thus, a delegatee colluding with the proxy cannot reveal Alice's private key. However, their scheme cannot achieve the revocation functionality. Therefore, achieving a revocable identity-based broadcast proxy re-encryption scheme is a challenging work.

3. PROPOSED WORK

We present our concrete construction for our scheme and further give the security proof of the proposed scheme.

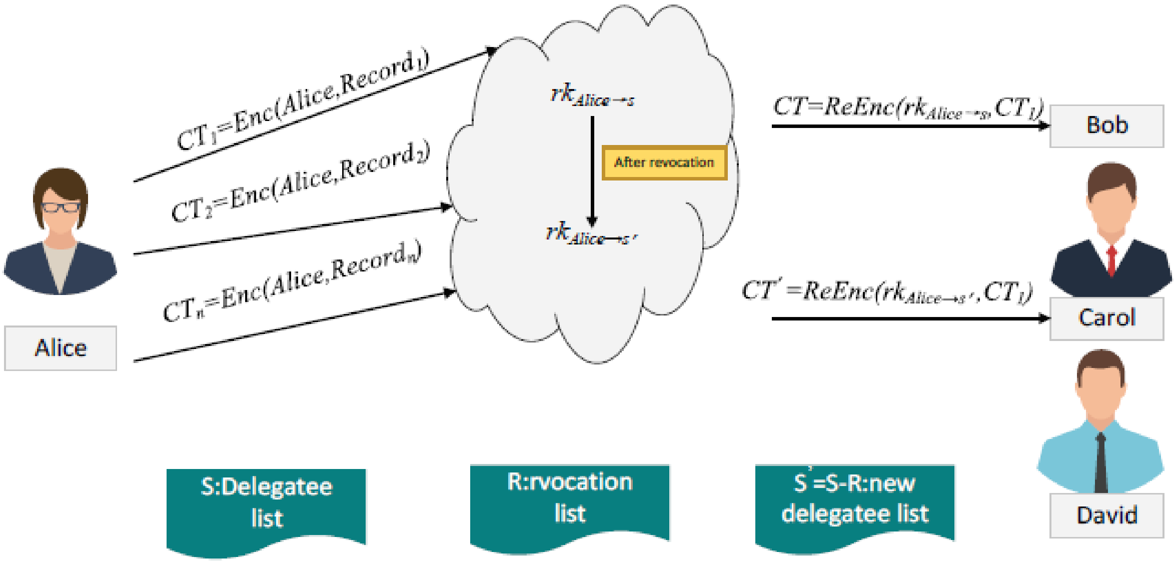
CT_1=Enc(Alice,Record$_1$)
CT_2=Enc(Alice,Record$_2$)
CT_n=Enc(Alice,Record$_n$)
$rk_{Alice \to s}$
After revocation
$rk_{Alice \to s'}$
CT=ReEnc($rk_{Alice \to s}$,CT_1)
CT'=ReEnc($rk_{Alice \to s'}$,CT_1)
Alice
Bob
Carol
David
S:Delegatee list
R:rvocation list
S'=S-R:new delegatee list

Fig. 1: RIB-BPRE System Model

This system model consists and implements the following modules:

Alice:

- In this module, Alice can register and login to the system and he upload the file data in to untrusted cloud server.
- Here, he can view the uploaded file details.

Bob:

- In this module, Bob can register and login to the system and he search the file data.
- Here, he/ she can view the file details and request the secrete key.
- After getting secrete key from cloud server he/she request the trapdoor.
- Finally, he can decrypt and download the file data.

Proxy Server:

- In this module, proxy server finds the user secrete key requests and provide the response.
- It can also find the trapdoor request to search the data and view the uploaded data files.

Cloud Server:

- In this module, cloud server login to the system and view all users' details, authorize them.
- Cloud provider can view the files uploaded to the system and he check the file decrypt and download requests and accept/reject the request.
- He can also generate the various reports.

Our scheme supports the following features:

• Alice encrypts each volunteer's genome data under her identity and sends the encrypted genome data to the proxy — the cloud server. She also maintains a list S of delegatees (her colleagues). For the proxy to run re-encryption, Alice computes a re-encryption key (rk_S) and shared rk_S with the cloud server. If the list S does not change, the proxy is able to re-encrypt the encrypted data from Alice as normal. Once the delegatees receive the data, they can decrypt it by their own private keys.

• One day, one of Alice's research fellows, Bob, decides to quit the job. Thus, the system must revoke Bob's access to the data because he is no longer an authorized staff. Then Alice can create a revocation list (R), update her delegatee list (S' = S −R) and notify the proxy there is a change to delegatee list. In this case, the proxy can re-generate the re-encryption key (rk_S') without knowing Alice's private key, which is the beauty of our RIB-BPRE scheme.

Figure-1 illustrates the idea of a RIB-BPRE system for medical research. In such a system, the user Alice herself maintains the delegatee revocation list. With the motivation in mind, we present a novel security notion — revocable identity-based broadcast proxy re-encryption

(RIB-BPRE). In the RIBBPRE scheme, the proxy can revoke a set of delegates, designated by the delegator, from the re-encryption key.

Algorithm

A revocable identity-based broadcast proxy re-encryption scheme consists of the following algorithms:

- **Setup (λ, N)** → (*mpk,msk*): The Setup algorithm is run by a trusted party, on input a security parameter λ and the maximum number N of receivers in one encryption. Outputs the master public parameters *mpk* and a master secret key *msk*.
- **Extract (*msk, id*)** → sk_{id}: The Extract algorithm is run by the trusted party to generate a private key for each identity. It takes as input the master secret key *msk*, and an identity id, outputs a private key skid.
- **Enc (*id*, M)** → C: The encryption algorithm Enc is run by anyone who encrypts the message with the delegator's identity. It takes as input a message M, an identity id, outputs the original ciphertext C that can be further re-encrypted.
- **RKeyGen (*id*, sk_{id}, S, k)** → *rk*: The RKeyGen algorithm is run by the delegator to generate a re-encryption key. It takes as input an identity id, private key skid, a set of delegates' identities S = {id_1,··· ,id_n} and a maximum revocation number k, where id / ∈ S and $k \leq n \leq N$. Outputs the re-encryption key *rk*. The re-encryption key *rk* can be used to convert an original ciphertext C under id to a new broadcast ciphertext CT under S.
- **Revoke (*rk*, S, R)** → rk_0: The Revoke algorithm is run by a proxy to generate a new re-encryption key that revokes identities from the sharing list. It takes as input a re-encryption key *rk* for identity set S, a revocation identity set R, where $R \subseteq S$ and $|R| \leq K$. Outputs a new re-encryption key rk_0. The re-encryption key rk0 can be used to convert an original ciphertext C under id to a new ciphertext CT under S −R.
- **ReEnc (C, *rk*)** → CT/⊥: The re-encryption algorithm ReEnc is run the proxy to transform the delegator's ciphertext to the delegatees' ciphertext. It takes as input an original ciphertext C, a re-encryption key *rk*, outputs the re-encrypted ciphertext CT or an error symbol ⊥.
- **Dec (sk_{id}, C/CT)** → m/⊥: The Dec algorithm is run by the delegator (or a delegatee) to decrypt the original ciphertext (or re-encrypted ciphertext). It takes as input a private key skid, an original/re-encrypted ciphertext C/CT. Outputs the plaintext M if the ciphertext is a valid ciphertext or an error symbol ⊥ otherwise. Note that, we omit the master public parameter mpk as other algorithms' input for the simplicity.

4. EXPERIMENTS & RESULTS

The comparison of computation cost between our scheme. Let Dec (Or) and Dec (Re) denote the decryption operation of an original ciphertext and re-encrypted ciphertext. We omit the computation cost of hash functions as it is much less than the computation of a bilinear paring and exponentiation in group.

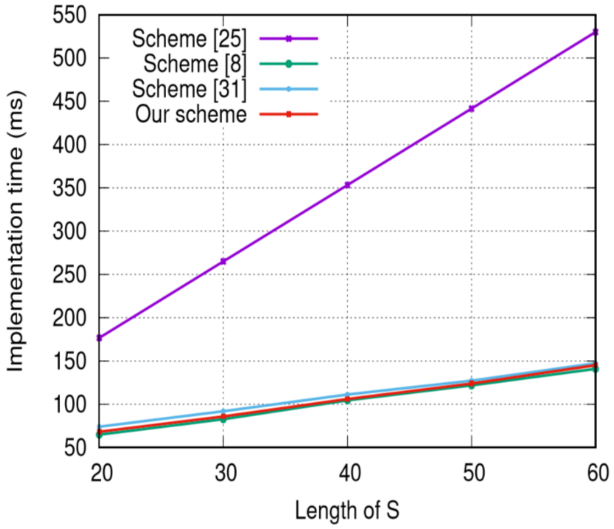

Fig. 2: RKeyGen Execute Time Comparison

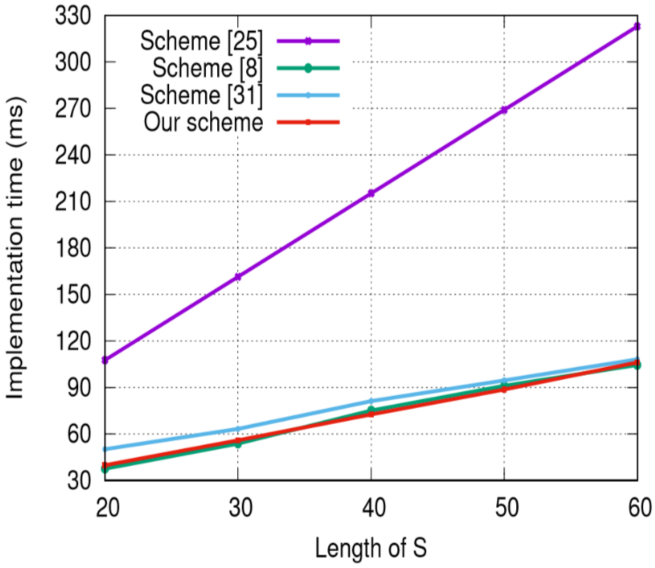

Fig. 3: Dec (Re) Execute Time Comparison

1) Execute Time: In our experiment we set the maximum size of the set of delegatees in one encryption N = 100. We varied |S| from 20 to 60 with step 10, and at the meanwhile varied k from 12 to 36 with step 6 and l from 10 to 30 with step 5. The execute time is summarized the execution time of the algorithms run by the data user and the proxy. We observe that the execution time of Extract, Enc, ReEnc, Revoke and Dec (Or) algorithms are almost constant.

While the execution time of RKeyGen and Dec (Re) algorithms are almost linear with the size of S.

2) Execute Time Comparison: In this experiment, we compare our scheme with [25] and [8] in RKeyGen and Dec (Re) algorithms as the execution time is linear with|S|. Further, we execute RKeyGen and Dec (Re) in [25] |S| times to achieve the same broadcast effect. The execute time comparison is showed in Fig.1 for RKeyGen algorithm and Fig.2 for Dec (Re) algorithm.

Figure 2 and Figure 3 show that, our scheme is almost as efficient as [8] and [31] in RkeyGen and Dec(Re) algorithms. However, our proposed scheme achieves the revocable functionality that is not provided in [8] and [31]. When compared with [25], our scheme is much more efficient, especially when |S| grows.

5. CONCLUSION

In this paper, we defined revocable identity-based broadcast proxy re-encryption, proposed a concrete construction under the definition and proved our scheme is CPA secure in the random oracle model. More importantly, the property and performance comparison reveal that our proposed scheme is efficient and practical. Furthermore, our RIB-BPRE scheme can nicely support key revocation for a data sensitive system in a cloud environment, for example, a volunteer-based genome research system. While this work has resolved the issue of key revocation for data sharing, it motivates some interesting open problems such designing RIB-BPRE scheme without random oracles and how to support more expressive on identities.

REFERENCES

[1] B. Dan and M. Franklin, “Identity-based encryption from the weil pairing,” in International Cryptology Conference, 2001, pp. 213–229.

[2] C. Cocks, “An identity-based encryption scheme based on quadratic residues,” in Cryptography and Coding, Ima International Conference, Cirencester, Uk, December, 2015, pp. 360–363.

[3] A. Sahai and B. Waters, “Fuzzy identity-based encryption,” in International Conference on Theory and Applications of Cryptographic Techniques, 2005, pp. 457–473.

[4] K. Liang and W. Susilo, “Searchable attribute-based mechanism with efficient data sharing for secure cloud storage,” IEEE Transactions on Information Forensics and Security, vol. 10, no. 9, pp. 1981–1992, 2017.

[5] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in International Conference on the Theory and Applications of Cryptographic Techniques, 1998, pp. 127–144.

[6] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in International Conference on Applied Cryptography and Network Security, 2007, pp. 288–306.

[7] C. K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional proxy broadcast re-encryption," Lecture Notes in Computer Science, vol. 5594, pp. 327–342, 2009.

[8] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email," IEEE Transactions on Computers, vol. 65, no. 1, pp. 66–79, 2015.

[9] S. Berkovits, "How to broadcast a secret," in International Conference on Theory and Application of Cryptographic Techniques, 1991, pp. 535– 541.

[10] A. Fiat and M. Naor, "Broadcast encryption," in International Cryptology Conference, 1993, pp. 480–491.

[11] J. Anzai, N. Matsuzaki, and T. Matsumoto, "A quick group key distribution scheme with efficient ntity revocation," Proc Asiacrypt, vol. 1716, pp. 333–347, 1999.

[12] D. Halevy and A. Shamir, "The lsd broadcast encryption scheme," in International Cryptology Conference on Advances in Cryptology, 2002, pp. 47–60.

[13] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," Crypto, vol. 2001, pp. 41–62, 2001.

[14] R. Sakai and J. Furukawa, "Identity-based broadcast encryption," Journal of Electronics and Information Technology, vol. 33, no. 4, pp. 1047– 1050, 2007.

[15] C. Delerabl, "Identity-based broadcast encryption with constant size ciphertexts and private keys," in Advances in Crypotology International Conference on Theory and Application of Cryptology and Information Security, 2007, pp. 200–215.

Author's Profile:

K. Venkateswarlu has received his MCA degree from Saraswathi Valu College of Engineering, Vellore affiliated to Anna University in 2010 and M.Tech degree in Computer Science from PBR Vits, Kavali affiliated to JNTU, Ananthapur in 2014 respectively. He is dedicated to teaching field from the last 6years. He has guided 25 P.G students. At present he is working as an Assistant Professor in Narayana Engineering College, Gudur, Andhra Pradesh, India.

B. Padmaja has Received her B.Sc Degree in Computer Science from D.R.W Degree College, Gudur affiliated to Vikrama Simhapuri University, Nellore in 2017 and pursuing PG Degree in Master of Computer Applications (M.C.A) from Narayana Engineering College, Gudur affiliated to JNTU Anantapur, Andhra Pradesh, India.