International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 4
ISSN: 2454-4248
15 – 17

Re-Encryption Scheme for Providing Data Security in Clouds

Namrata Khadtar
Department of Computer Engineering
Datta Meghe College of Engineering
Airoli, India.
e-mail: namrata.khadtar@gmail.com

Urmila Mahale
Department of Computer Engineering
DattaMeghe College of Engineering
Airoli, India.
e-mail:urmilamahale95@gmail.com

Pooja Singh Dohare
Department of Computer Engineering
DattaMeghe College of Engineering
Airoli, India
e-mail: poojasd97@gmail.com

Prof. Prashant Itankar
Department of Computer Engineering
DattaMeghe College of Engineering
Airoli, India.
e-mail: ittu_prashant@yahoo.co.in

Abstract:- Cloud computing is the trendy topic all over the world. With increase in popularity of cloud computing, more and more enterprises will outsource their sensitive data for sharing in a cloud. Cloud computing allows the users to share the data among the members of cloud. One of the issue in cloud computing is data security. Here the problem is whenever a user is revoked from cloud the data owners will send re-encryption command to cloud in order to re-encrypt the data so that the data is prevented from revoked users. In this paper, we propose a time-based re-encryption scheme using blowfish algorithm. This scheme allows the cloud to automatically re-encrypt the data based on the internal clock, users can access data within given time period, after time period get over user cannot access data, this scheme also prevents the revoked users from decrypting the data using their old decryption keys.

Keyword*: Cloud security, Blowfish, Re-encryption.*

*****

I. INTRODUCTION

In today's world cloud computing becomes the buzz in the market as it provides various services.It is important to secure the cloud, one of the technique to secure cloud is store encrypted data in cloud. The problem of storing encrypted data in the cloud lies in revoking access rights from users. A user whose is revoked will still retain the keys issued earlier, and thus can still decrypt data using old keys.

A solution for this is to let the data owner immediately re-encrypt the data, so that the revoked users cannot decrypt the data using their old keys, while distributing the new keys to the remaining authorized users. Due to frequent revocation this solution will lead to a performance bottleneck.

An another solution is to apply the proxy re-encryption (PRE) technique. This approach is also

called command-driven re-encryption scheme, where cloud servers execute re-encryption while receiving the re-encryption commands from the data owner.

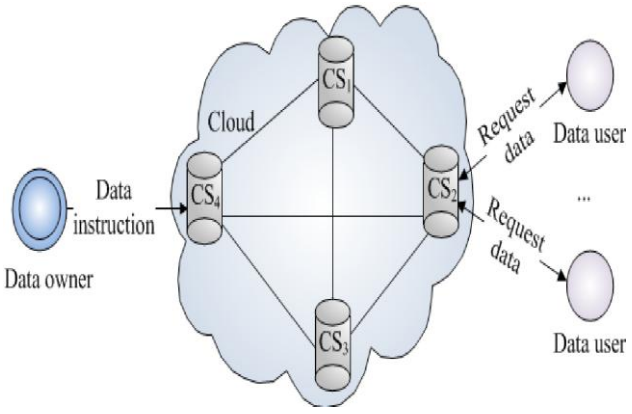

Fig. 1 A typical cloud environment

A cloud is large scale distributed system where a data owner's data is replicated over multiple servers. As a distributed system, the cloud will experience failures common to such systems, such as server crashes and network outages. As a result, re-encryption commands sent by the data owner may not receive by all the servers in a timely fashion, thus creating security risks.

 ISSN: 2454-4248
15 – 17

To illustrate, let us consider a cloud environment shown in Fig. 1, where the data owner's data is stored on cloud servers CS1, CS2, CS3, CS4. Assume that the data owner issues to CS4 a re-encryption command, which should be propagated to CS1, CS2, CS3. Due to a network outage, CS2 did not receive the command, and did not re-encrypt the data. At this time, if revoked users query CS2, they can obtain the old cipher text, and can decrypt it using their old keys. A better solution is to allow each cloud server to independently re-encrypt data without receiving any command from the data owner. In this paper, we propose a reliable re-encryption scheme in unreliable clouds. R3 is a time-based re-encryption scheme, which allows each cloud server to automatically re-encrypt data based on its internal clock.

II. LITERATURE SURVEY

S. Kamara and K. Lauter [1] Researchers have proposed storing encrypted data in the cloud to defend against the CSP. Under this approach, users are revoked by having a third party to re-encrypt data such that previous keys can no longer decrypt any data. The solution by...for instance, lets the data owner issue a re-encryption key to an untrusted server to re-encrypt the data. Their solution utilizes PRE [4], which allows the server to re-encrypt the stored cipher text to a different cipher-text that can only be decrypted using a different key. During the process, the server does not learn the contents of the cipher text or the decryption keys. ABE is a new cryptographic technique that efficiently supports fine grained access control.

Guojun Wang, Qin Liu, JieWub, M. Guo [2]

proposed a hierarchical attribute-based encryption

scheme (HABE) by combining a hierarchical identity based encryption (HIBE) scheme and a cipher text policy attribute-based encryption (CP-ABE) scheme.

J. Bettencourt, A. Sahai, and B. Waters [3] introduced attribute-based encryption (ABE) as a new means for encrypted access control. Attribute-based encryption (ABE) is a public-key algorithm based one to many encryptions that allows users to encrypt and decrypt data based on user attributes. In their context, the role of the parties is taken by the attributes. Thus, the access structure will contain the authorized sets of attributes. They restrict the attention to monotone access structures. However, it is also possible to realize general access structures using the techniques by having the attribute as a separate attribute altogether. Thus, the number of attributes in the system will be doubled. From now on, unless stated otherwise, by an access structure we mean a monotone access structure.

III. EXISTING SYSTEM

The main problem of storing encrypted data in the cloud system lies in *revoking* access rights from users. A user whose permission is revoked will still be able to get the keys issued earlier, and thus can decrypt data in the cloud. A solution for this problem is to let the data owner immediately re-encrypt the data, so that the revoked users cannot decrypt the data using their old keys, while distributing the new keys to the remaining authorized users. The above solution will lead to a performance bottleneck, especially when there are frequent user revocations. An alternative solution is to apply the *proxy re-encryption* (PRE) technique. It takes advantage of the abundant resources in a cloud by delegating the cloud to re-encrypt data. This approach is also called command-driven re-encryption scheme, where cloud servers execute re-encryption while receiving commands from the data owner. However, command-driven re-encryption schemes do not consider the underlying system architecture of the cloud environment.

IV. PROPOSED SYSTEM

We propose a *reliable re-encryption scheme in clouds* (R3 scheme for short). R3 is a time-based re-encryption scheme, and it allows each cloud server to automatically re-encrypt data based on its internal clock. The basic idea of the R3 scheme is to associate the data with an *access control* and an *access time*. Unlike the command-driven re-encryption scheme, the data owner and the CSP share a secret key, from which each cloud server will be able to re-encrypt data by updating the data access time according to its own internal clock. Even through the R3 scheme relies on *time*, it does not require perfect clock synchronization among cloud servers. Classical clock synchronization techniques that ensure loose clock synchronized [5]- [8]in the cloud are sufficient. The main contributions are as follows:

1) We propose an automatic, time-based, Re-encryption scheme suitable for cloud environments with unpredictable server crashes and network outages.
2) We are using Blowfish algorithm to perform re-encryption.
3) Our solution does not require perfect clock synchronization among all of the cloud servers to maintain correctness.

V. BLOWFISH ALGORITHM

Blowfish is a symmetric block encryption algorithm which encrypts block data of 64-bits at a time. It is designed in consideration with fast, compact, simple and secure. It

 ISSN: 2454-4248
15 – 17

follows the feistel network and this algorithm is divided into two parts.

1.**Key-expansion**: It will convert a key of at most 448 bits into several sub key arrays totaling 4168 bytes. Blowfish uses large number of sub keys. These keys are generating earlier to any data encryption or decryption.

2. **Data Encryption**: It is having a function to iterate 16 times of network. Each round consists of key-dependent permutation and a key and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookup tables for each round.

Algorithm: Blowfish Encryption

Divide x into two 32-bit halves: xL, xR
For i = 1 to 16:
xL = XL XOR Pi
xR = F(XL) XOR xR
Swap XL and xR
Swap XL and xR (Undo the last swap.)
xR = xR XOR P17
xL = xL XOR P18
Recombine xL and xR.

VI. CONCLUSION AND FUTURE MODIFICATION

We proposed the R3 scheme, used for managing access controlbased on the cloud server's Internal clock. Our technique does not rely on the cloud to reliably propagate re-encryption commands to all servers to ensure access control correctness.

We showed that our solutions remain secure without perfect clock synchronization so long as we can bind the time difference between the servers and the data owner.

The future enhancement to this project is planned such that a possible improvement is to let the owner issue a valid user a special seed value which the user can then use to generate keyson his own. Our solution can be extended to allow users to perform data updates in addition to data owners.

REFERENCES:

[1] S. Kamara and K. Lauter, "Cryptographic cloud storage," Financial Cryptography and Data Security, 2010.
[2] Guojun Wang, Qin Liu, JieWub, Minyi Guo," Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers", Jul 1, 2011
[3] J. Bettencourt, A. Sahai, and B. Waters " Ciphertext-Policy Attribute Based Encryption "in Proceedings of IEEE Symposium on Security and Privacy, pp. 321V334, 2007
[4] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," Advances in Cryptology– EUROCRYPT, 1998.
[5] F. Cristian, "Probabilistic clock synchronization," Distributed Computing, 1989.
[6] K. Romer, "Time synchronization in ad hoc networks," in Proc. of ACM MobiHoc, 2001.
[7] P. Ramanathan, K. Shin, and R. Butler, "Fault-tolerant clock synchronization in distributed systems," Computer, 2002.
[8] N. Antonopoulos and L. Gillam, "Cloud Computing: Principles, Systems and Applications," Springer Publishing Company, 2010.