International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 2, Issue 1, January 2015, PP 18-21
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
www.arcjournals.org

Data Security for Unreliable Clouds Using Reliable Encryption

Veeresh[1], K. Arjun[2]

[1]PG Scholar, CSE, BITS, Adoni AP, India
[2]Asst Professor, CSE, BITS, Adoni AP, India

Abstract: *A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data are leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). We propose data allocation strategies that improve the probability of identifying leakages. In Image and Video we can apply watermarking for detecting guilty agent. All the data will be stored on the cloud so Reliable Re-encryption in unreliable clouds technique is used. In this data owner store encrypted data in the cloud, and issue decryption keys to authorized users. When a user is revoked, the data owner will issue re-encryption commands to the cloud to re-encrypt the data, to prevent the revoked user from decrypting the data, and to generate new decryption keys to valid users, so that they can continue to access the data. In this project a time based re-encryption scheme is used for cloud servers to automatically re-encrypt data based on their internal clocks.*

Keywords: *Data Leakage, Watermarking, Re-Encryption, Cloud, Guilty agent*

1. INTRODUCTION

In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. For example, a hospital may give patient records to researchers who will devise new treatments. Another enterprise may outsource its data processing, so data must be given to various other companies. We call the owner of the data the distributor and the supposedly trusted third parties the agents. Our goal is to identify the agent that leaked the Distributor's sensitive data. We consider applications where the original sensitive data cannot be perturbed. For example, one can add random noise to certain attributes, or one can replace exact values by ranges [1].Traditionally, leakage detection is handled by watermarking,. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious. In this paper, we study unobtrusive techniques for detecting leakage of a set of objects or records. If the distributor sees "enough evidence" that an agent leaked data, he may stop doing business with him, or may initiate legal proceedings. In this paper, we develop a model for assessing the "guilt" of agents. We also consider the option of adding "fake" objects to the distributed set. Such objects do not correspond to real entities but appear realistic to the agents. In a sense, the fake objects act as a type of watermark for the entire set, without modifying any individual members. If it turns out that an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty. One technique to protect the data from a possible untrusted CSP is for the data owner to encrypt the outsourced data [2],[3]. Flexible encryption schemes such as attribute based encryption (ABE) [4]-[6] can be used. ABE allows data to be encrypted using an access structure comprised of different attributes. For example, a file encrypted using the access structure $\{(\alpha 1 \wedge \alpha 2) \vee \alpha 3\}$ means that either a user with attributes $\alpha 1$ and $\alpha 2$, or a user with attribute $\alpha 3$, can decrypt the file. A user whose permission is revoked still decrypt data in the cloud. A naive solution 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. This 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 [7],[8]. This approach takes advantage of the abundant resources in a cloud by delegating the cloud to re-encrypt data, where cloud servers execute re-encryption while receiving commands from the data owner. 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 propagate to all of the servers in a timely fashion, thus creating security risks.

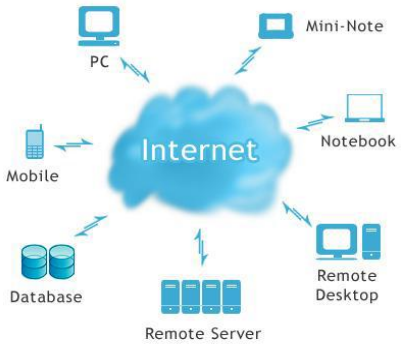

Fig1. *A typical cloud environment*

2. SYSTEM ARCHITECTURE

Cloud computing is an emerging computing paradigm in which resources of the computing infrastructure are provided as services over the Internet. As promising as it is, this paradigm also brings forth many new challenges for data security and access control when users outsource sensitive data for sharing on cloud servers, which are not within the same trusted domain as data owners. To keep sensitive user data confidential against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. A cloud is essentially a large scale distributed system where a data owner's data is replicated over multiple servers for high availability. 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 propagate to all of the servers in a timely fashion, thus creating security risks.

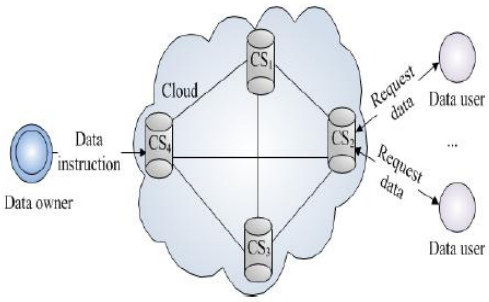

Fig2. *The cloud architecture with data owner, cloud service provider and data user*

To illustrate, let us consider a cloud environment shown in Fig. 2, where the data owner's data is stored on cloud servers CS1, CS2, CS3, and CS4. Assume that the data owner issues to CS4 a re-encryption command, which should be propagated to CS1, CS2, and 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. The disadvantage was Bulk Data Transfers- Bringing a lot of data into or out of a cloud instance takes a good deal of time. Without a high-capacity connection, it could take days to load all that data. So in this paper we propose a reliable re-encryption scheme in unreliable clouds (R3 scheme for short). R3 is a time-based re-encryption scheme; this 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.

- We propose an automatic, instant mailing system suitable for cloud environments.
- We extend an ABE scheme by advanced mailing password to the valid data user

- Our solution does not require perfect clock synchronization among all of the cloud servers to maintain correctness.

The advantage of this R3 scheme

- When you use internet with the cloud services then your company will have lots more room to store the files and that they need to store.
- User identified the data losses.
- Data security and access control when users require data for sharing on cloud server.

3. PROPOSED SYSTEM

Our goal is to detect when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data.

Perturbation is a very useful technique where the data is modified and made "less sensitive" before being handed to agents. We develop unobtrusive techniques for detecting leakage of a set of objects or records. In this section we develop a model for assessing the "guilt" of agents. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker. Finally, we also consider the option of adding "fake" objects to the distributed set. Such objects do not correspond to real entities but appear realistic to the agents. In a sense, the fake objects acts as a type of watermark for the entire set, without modifying any individual members. If it turns out an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty. We also propose a reliable re-encryption scheme in unreliable clouds (R3 scheme for short). R3 is a time-based re-encryption scheme, which 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. Each user is issued keys associated with attributes and attribute effective times. The data can be decrypted by the users using the keys with attributes satisfying the access control, and attribute effective times satisfying the access time. Unlike the command-driven re-encryption scheme, the data owner and the CSP share a secret key, with which each cloud server can 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 in the cloud are sufficient. The main contributions are as follows

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

4. RELATED WORK

Researchers have proposed storing encrypted data in the cloud to defend against the CSP [1], [2]. 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 [14]–[16]. The solution by [15] for instance, lets the data owner issue a re-encryption key to an untrusted server to re-encrypt the data. Their solution utilizes PRE [6], 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. The combination of PRE and ABE was first introduced by [9], and extended by [8], [17]. In [8], a hierarchical attribute-based encryption (HABE) scheme is proposed to achieve high performance and full delegation. The main difference between prior work and ours is that we do not require the underlying cloud infrastructure to be reliable in order to ensure correctness. Our scheme relies on time to re-encrypt data. However, in a cloud, the internal clock of each cloud server may differ. There have been several solutions to this problem. For instance, [10] proposed a probabilistic synchronization scheme, which exchanges messages to get remote servers' accurate clocks with high probability.

At [11] used message delay to estimate the maximal difference between two communicating nodes to synchronize the clocks. At [13] proposed a clock synchronization scheme for cloud environments. At [14] the encryption and re-encryption is done using ABE technique. In this paper we have used instant mailing scheme for instant password.

5. CONCLUSION AND FUTURE ENHANCEMENT

The proposed R3 scheme, a new method for managing access control based 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 using instant mailing system. The future enhancement to this project is planned such that when data user request for a file if he is valid user for easy access the password is sent to the valid data user mobile number as text message.

REFERENCES

[1] S. Kamara and K. Lauter, "Cryptographic cloud storage," Financial Cryptography and Data Security, 2010.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," Communications of the ACM, 2010.

[3] A. Sahai and B. Waters, "Fuzzy identity-based encryption," Advances in Cryptology–EUROCRYPT, 2005.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. of ACM CCS, 2006.

[5] J. Bethencourt, A. Sahai, and B. Waters, "Cipher text-policy attribute based encryption," in Proc. of IEEE Symposium on S&P, 2007.

[6] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," Advances in Cryptology–EUROCRYPT, 1998.

[7] A. Boldyreva, V. Goyal, and V. Kumar, "Identity-based encryption with efficient revocation," in Proc. of ACM CCS, 2008.

[8] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in Proc. of ACM CCS (Poster), 2010.

[9] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in Proc. of IEEE INFOCOM, 2010.

[10] F. Cristian, "Probabilistic clock synchronization," Distributed Computing, 1989

[11] K. Romer, "Time synchronization in ad hoc networks," in Proc. of ACM MobiHoc, 2001.

[12] P. Ramanathan, K. Shin, and R. Butler, "Fault-tolerant clock synchronization in distributed systems," Computer, 2002.

[13] N. Antonopoulos and L. Gillam, "Cloud Computing: Principles, Systems and Applications," Springer Publishing Company, 2010.

[14] Qin Liu, Chiu C. Tan, Jie Wu, and Guojun Wang, "Reliable Re- Encryption In Unreliable Clouds"IEEE,2011.