Reliable Data in Cloud Using Re-Encryption

Supriya Prajapati, Neha Gulhane, Pranav Betkekar, Prof. B.W.Balkhande

Computer Engineering, Bharati Vidyapeeth College of Engineering, Mumbai University, Navi Mumbai, India.
Computer Engineering, Bharati Vidyapeeth College of Engineering, Mumbai University, Navi Mumbai, India.
Computer Engineering, Bharati Vidyapeeth College of Engineering, Mumbai University, Navi Mumbai, India.
Computer Engineering, Bharati Vidyapeeth College of Engineering, Mumbai University, Navi Mumbai, 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.

Index Terms- Data Leakage, Watermarking, Re-Encryption, Cloud, Guilty agent

I. 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 α1 and α2, or a user with attribute α3, 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.

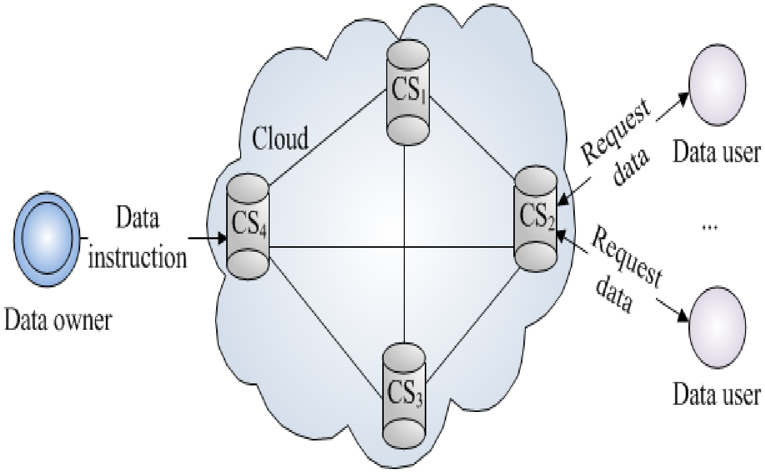


Fig. 1.1 A typical cloud environment

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.

II. 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.

III. IMPLEMENTATION OF CLOUD

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

IV. DATA LEAKAGE DETECTION AND CLOUD MODULES

Data leakage detection modules:

1.Data Allocation Module:

The main focus of our project is the data allocation problem as how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent, Admin can send the files to the authenticated user, users can edit their account details etc. Agent views the secret key details through mail. In order to increase the chances of detecting agents that leak data.

2. Fake Object Module:

The distributor creates and adds fake objects to the data that he distributes to agents. Fake objects are objects generated by the distributor in order to increase the chances of detecting agents that leak data. The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. Our use of fake objects is inspired by the use of "trace" records in mailing lists. In case we give the wrong secret key to download the file, the duplicate file is opened, and that fake details also send the mail. Ex: The fake object details will display.

3. Optimization Module:

The Optimization Module is the distributor's data allocation to agents has one constraint and one objective. The agent's constraint is to satisfy distributor's requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. His objective is to be able to detect an agent who leaks any portion of his data. User can be able to lock and unlock the files for secure.

4. Data Distributor:

A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data is leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. Admin can be able to view the which file is leaking and fake user's details also.

5.Image & Video Watermarking:

Watermarking is a technique used to hide data or identifying information within digital multimedia With the advent of internet, creation and delivery of digital data (images, video and audio files) has grown many fold. With this, issues like protection of rights of the content and proving ownership arises. Digital image watermarking came as a technique and a tool to overcome shortcomings of current copyright laws for digital data .

Cloud modules:

1.Intuition:

The data owner will first generate a shared secret key to the CSP. Then, after the data owner encrypts each file with the appropriate attribute structure and time slice, the data owner uploads the file in the cloud. The CSP will replicate the file to various cloud servers. Each cloud server will have a copy of the shared secret key.

2. Data owner initialization:

The data owner runs the Setup function to initiate the system. When the data owner wants to upload file F to the cloud server, it first defines an access control A for F, and then determines the current time slice TSi. Finally, it runs the Encrypt function with A and TSi to output the ciphertext. When the data owner wants to grant a set of attributes in a period of time to data user Alice, it runs the GenKeyfunction with attributes and effective times to generate keys for Alice.

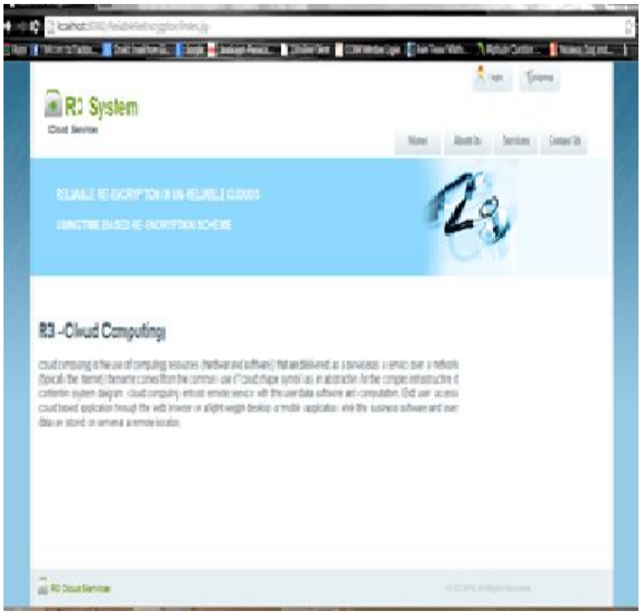
3.Data user read data:

When data user Alice wants to access file F at TSi, she sends a read command R(F) to the cloud server, where F is the file name. On receiving the read command R(F), the cloud server runs the ReEncryptfunction to re-encrypt the file with TSi. On receiving the ciphertext, Alice runs the Decrypt function using keys satisfying A and TSito recover F.

4.Attribute based encryption (ABE):

ABE allows data to be encrypted using an access structure comprised of different attributes. Instead of specific decryption keys for specific files, users are issued attribute keys. Users must have the necessary attributes that satisfy the access structure in order to decrypt a file.

V. GUI



VI. DESIGN AND IMPLEMENTATION

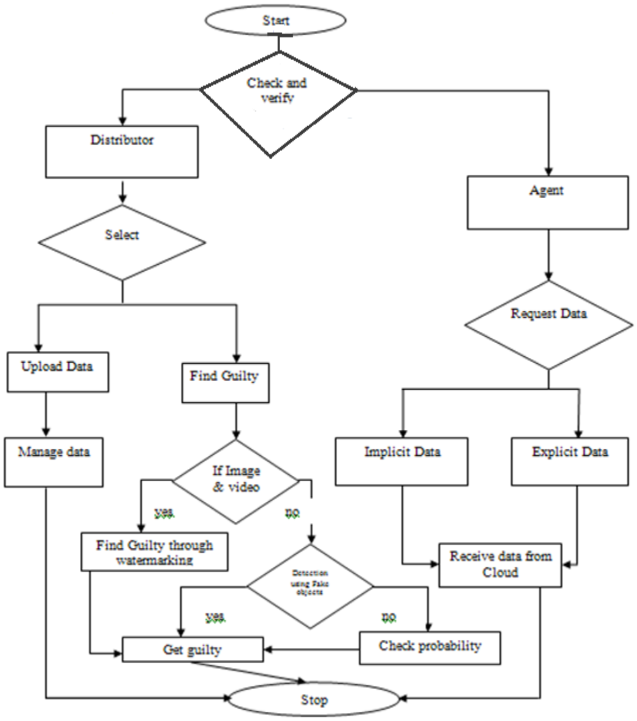


Fig 1.2 Flow chart

VII. CONCLUSION

Finds the probability that an agent is responsible for a leak. We have shown that distributing objects properly can make a significant difference in identifying guilty agents, especially in cases where there is large overlap in the data that agents must receive. We proposed the 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 without perfect clock synchronization so long as we can bound the time difference between the servers and the data owner.

REFERENCES

[1] L. Sweeney, "Achieving K-Anonymity Privacy Protection Using Generalization and Suppression," http://en.scientificcommons. org/43196131, 2002.

[2] R. Agrawal and J. Kiernan, "Watermarking Relational Databases," Proc. 28th Int'l Conf. Very Large Data Bases (VLDB '02), VLDB Endowment, pp. 155-166, 2002.

[3] P. Bonatti, S.D.C. di Vimercati, and P. Samarati, "An Algebra for Composing Access Control Policies," ACM Trans. Information and System Security, vol. 5, no. 1, pp. 1-35, 2002.

[4] P. Buneman, S. Khanna, and W.C. Tan, "Why and Where: A Characterization of Data Provenance," Proc. Eighth Int'l Conf. Database Theory (ICDT '01), J.V. den Bussche and V. Vianu, eds., pp. 316-330, Jan. 2001.

[5] P. Buneman and W.-C.Tan, "Provenance in Databases," Proc. ACM SIGMOD, pp. 1171-1173, 2007.

[6] Y. Cui and J. Widom, "Lineage Tracing for General Data Warehouse Transformations," The VLDB J., vol. 12, pp. 41-58, 2003.

[7] S. Czerwinski, R. Fromm, and T. Hodes, "Digital Music Distribution and Audio Watermarking," http://www.scientificcommons.org/43025658, 2007.

[8] F. Guo, J. Wang, Z. Zhang, X. Ye, and D. Li, "An Improved Algorithm to Watermark Numeric Relational Data," Information Security Applications, pp. 138-149, Springer, 2006.

AUTHORS

First Author – Supriya Prajapati, Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, supriya.prajapati@outlook.com.
Second Author – Neha Gulhane, Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, neha.gulhane@outlook.com.
Third Author – Pranav Betkekar, Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, pranavbetkekar_cyborg@yahoo.com.
Forth Author – Prof. B.W.Balkhande, Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, balabalkhande@hotmail.com