ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

Secure file sharing technique on cloud storage using an aggregate-key

Mr. Sharan L Pais[1], Mr. Shrikanth N G[2]

M.Tech., CSE, SDIT, Mangalore, India[1]

Assistant Professor, Department of CSE, SDIT, Mangalore, India[2]

Abstract: An important functionality of cloud storage is data sharing. Secure and efficient data sharing is the main aspect. Cloud storage can provide strong protection, good reliability and accessibility, disaster recovery, and lowest cost. On cloud storage anyone can share data as much they want to i.e. only selected content can be shared. Cryptography helps the data owner to share the data in a safer way. So user encrypts data and uploads on server. Aggregate key cryptosystem produces a constant size key such that efficient delegation of decryption rights for any file is possible in a secure manner. Here a public-key cryptosystem is used, which generates a public key which is used to decrypt the file. But the public is aggregated with one more key. The difference is one can collect a set of secret keys and make them as small size as a single key with holding the same ability to decrypt the file securely. This compact aggregate key can be efficiently sent to the clients or to be stored in a smart card with little secure storage.

Keywords: Aggregate key, data sharing, Encryption, Decryption, secret key.

I. INTRODUCTION

Cloud technology is the emerging trend. Cloud storage is nowadays very much popular and has great demand. Cloud storage is storing of data to the physical storage which is maintained by third party. Cloud storage is saving of digital data in logical pool and physical storage spans multiple servers which are manage by third party. Its flexible and cost optimizing characteristic motivates the end user as well as enterprises to store the data on cloud. The insider attack is one of security concern which needs to be focused.

Third party is responsible for keeping data available and accessible and physical environment should be protected and running at all time. Instead of storing data on a local storage, we save data to remote storage which is accessible from anywhere and anytime. It provides more mobility to data.

While considering data privacy, we cannot rely upon traditional technique of authentication, because unexpected privilege escalation will expose all data. Solution is to encrypt data before uploading to the server with uploader's own key. Data sharing is again important functionality of cloud storage, because user can share data from anywhere and anytime and to anyone.

For example, organization may grant permission to access part of sensitive data to their employees. But challenge is that how to share encrypted data. Traditional way is user can download the encrypted data from storage, decrypt that data and send it to share with others, but it loses the importance of cloud storage.

There are two main Cryptographic techniques that can be applied- symmetric key encryption and asymmetric key encryption. In symmetric key encryption, same keys are used for encryption and decryption.

By contrast, in asymmetric key encryption different keys are used, public key for encryption and private key for decryption.

Suppose Alice put all data on Dropbox or any other cloud storage and she does not want to expose her data to everyone.

To secure her data she wants to encrypt the data before upload. If Bob ask her to share some data then Alice use share application of Dropbox[1].

But problem is how to share encrypted data? There are two severe ways:

- Alice encrypts data with single secret key and shares that secret key directly with the Bob.
- Alice can encrypt data with distinct keys and send Bob corresponding keys to Bob via secure channel.

In first approach, unwanted data also get expose to the Bob, which is inadequate.

In second approach, number of keys is as many as number of shared files, which may be hundred or thousand as well as transferring these keys require a secure channel and storage space which can be more expensive.

Therefore best solution to above problem is Alice encrypts data with distinct public keys, and sends a single decryption key of constant size to Bob [1].

Since the decryption key should be sent via secure channel and kept secret using smart cards.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

1. http://www.dropbox.com

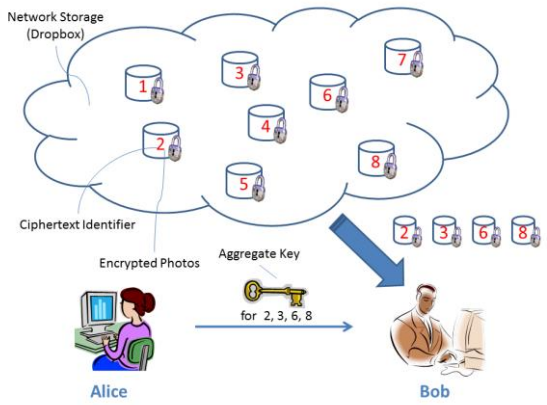

Fig1. Alice shares files with identifiers 2, 3, 6 and 8 with Bob by sending him an aggregate key

II. RELATED WORK

Cloud computing is one of the emerging field of study. So many experiments are carried out on data sharing and most of the techniques based on the encrypting and decrypting the data which is shared, because most important factor here to consider about is data security. So many techniques were used like,

A. Identity Based Encryption:

The beauty of Identity based encryption (IBE)(e.g. :[4], [5]) lies in the convenience of public key handling, in the sense that any identification such as email address, name or an IP number can serve as a public key of a party. However, such convenience is not inherited by a system where a party possesses many identities (e.g.: many Email IDs) and has to use them as his public keys. When his system is handled with a standard IBE, the user must manage all the private keys that are associated with all the public keys. However keeping these private keys is inconvenient to the user. Here a novel method is used where a private key can map to multiple public keys, means, we can use a private key to decrypt multiple ciphertexts; each was encrypted with different public key (identity).

B. Attribute-Based Encryption:

Another cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE) [6], [7]. In this cryptosystem, ciphertexts are labelled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt. For example, with the secret key for the policy $(1 \vee 4 \vee 6 \vee 9)$, one can decrypt ciphertext tagged with class 1, 4, 6 or 9. However, the main concern in ABE is collusion-resistance but not the compactness of secret keys used.

C. Compact Key in Symmetric-Key Encryption:

This scheme is motivated by the problem of supporting flexible Hierarchy in decryption power delegation. This approach achieves similar properties and performances as our scheme. However, it is designed for the symmetric-key settings instead [2], [3]. The encryptor needs to obtain the corresponding secret keys to encrypt data, which is not suitable for many of the applications. Since their method is used to generate a secret value rather than a pair of public/secret keys, it is unclear how to apply this idea for public-key encryption scheme.

III. AGGREGATE KEY CRYPTOSYSTEM

The main design of this scheme is concentrated on the security of the data shared on cloud. The proper authentication for cloud access is also a key issue. Here we made use of Dropbox as the third party cloud vendor. Initially the files are uploaded to the cloud space by the data owner. The essential encryption standards are used to protect the data privacy. The client, who wants to access the data or file must be a registered user of the system. The aggregate key is generated at the time of file sharing. Let's considers the different phases of working of the system. Setup Phase: This module is executed by the data owner. Here the admin or data owner will choose a public key, say pk for file upload. Encrypt (pk, m): This module is executed by the data owner. Here the file which is to be uploaded to the cloud space is encrypted using the public key, pk. Upload: The encrypted file is then uploaded to the cloud space. While uploading the authentication for access privileges are verified. So the data is safe in the encrypted format. Key sharing: The admin will share the file with sending the key to the clients. They should be already registered in the system. Key details are sent via a secure Email.

Decrypt: The client will extract the file sharing details from Email. After entering the details the access rights to the cloud space is verified and file is downloaded. File is decrypted using the public key.

IV. SHARING THE ENCRYPTED DATA

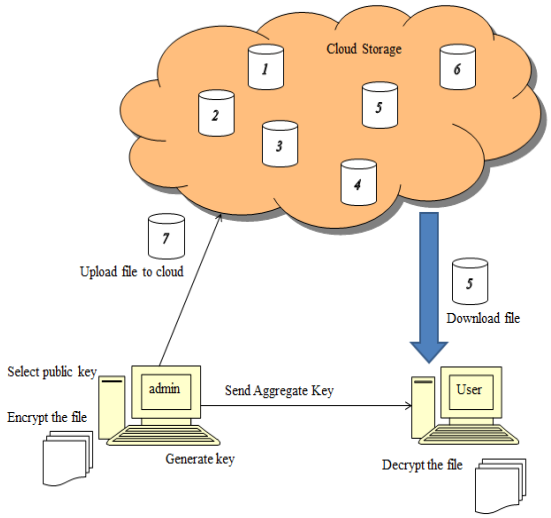

Fig 2. Data sharing using aggregate key

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

Consider that admin has many files under him which may be shared under his authority with the user or clients.

First the admin has to save this file in the cloud space so that it will be available to users at any time. So he will generate a public key, pk which is used to encrypt the file. Then he chooses the file to upload and it is encrypted using pk. After encryption the file is uploaded to the cloud space.

When the user needs to access the file, the admin will share the file details with the user. The aggregate key generated at this time. Then the details are sent via secure Email to the user. When the user gets the Email from the admin, he will get the file details. He can now enter the file name and key to download it, in his system. After downloading the decryption is carried out by extracting the key, pk from aggregate key. Then the file is saved in the predefined folder in the client system.

CONCLUSION

Flexible data sharing is vital thing in cloud computing. Users prefer to upload their data on cloud and among different users. Outsourcing of data to third party may lead to leak the private data of user to everyone. Encryption is a one solution which provides to share selected data with desired candidate. Sharing of decryption keys in secure way plays important role. Aggregate key cryptosystems provides delegation of secret keys for different files in cloud storage. The delegatee gets securely an aggregate key of constant size. It is required to improve the performance by allowing sharing of more number of files as the future enhancement.

REFERENCES

[1]. Cheng-Kang Chu ,Chow, S.S.M, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng , ―Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage‖, IEEE Transactions on Parallel and Distributed Systems. Volume: 25, Issue: 2. Year :2014.

[2]. J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records", in Proceedings of ACM Workshop on Cloud Computing Security (CCSW '09). ACM, 2009, pp. 103–114.

[3]. J. Benaloh, ―Key Compression and Its Application to Digital Fingerprinting,‖ Microsoft Research, Tech. Rep., 2009.

[4]. D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," in Proceedings of Advances in Cryptology – CRYPTO '01, ser. LNCS, vol. 2139. Springer, 2001, pp. 213–229.

[5]. A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in Proceedings of Advances in Cryptology - EUROCRYPT '05, ser. LNCS, vol. 3494. Springer, 2005, pp. 457–473.

[6]. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data," in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06). ACM, 2006, pp. 89–98.

[7]. M. Chase and S. S. M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," in ACM Conference on Computer and Communications Security, 2009, pp. 121–130.

[8]. D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," in Proceedings of Advances in Cryptology - CRYPTO '05, ser. LNCS, vol. 3621. Springer, 2005, pp. 258–275.