Re-encryption based Key Management with De-duplication Mechanism for Cloud

Anuja Phapale
AISSMSInstitute of InformationTechnology,Pune
Departmentof InformationTechnology

Akshat Vaidya
AISSMSInstitute of InformationTechnology,Pune
Departmentof InformationTechnology

Debashish Dwivedi
AISSMSInstitute of InformationTechnology,Pune
Departmentof InformationTechnology

Mayur Phanse
AISSMSInstitute of InformationTechnology,Pune
Departmentof InformationTechnology

ParthBhimani
AISSMSInstitute of InformationTechnology,Pune
Departmentof InformationTechnology

ABSTRACT

Cloud computing has many strong economic advantages, but clients are reluctant to trust a third-party cloud provider. To confront these security concerns, data can be transmitted and stored in encrypted form. There are challenges regarding the conditions of the generation, distribution, and usage of encryption keys in cloud systems, such as the safe place of keys. These idiosyncrasies lead to difficulties in achieving effective and scalable key management. In this work, a model for key management based on the principle of dynamic data re-encryption is practiced to a cloud computing system in a unique way to address the demands of a cloud environment. The proposed model is highly scalable, secure and efficient in a cloud computing reference, as keys are handled by the client for trust reasons, resource-intensive data re-encryption is handled by the cloud provider, and key distribution is minimized to conserve communication costs on. Attribute-based encryption is proposed to allow users access to cloud based on the satisfaction of required characteristics such that the higher computation load from cryptographic operations is assigned to the cloud provider. A versioning history mechanism effectively manages keys for a constantly changing user population and cross checks the session-ID of user. Furthermore a data de-duplication mechanism is added in order to allow efficient storage in cloud scenarios. Finally, an implementation on commercial cloud platform is used to validate the performance of the model.

General Terms

Cloud computing, security, key management, de-duplication, attribute based encryption

Keywords

Re-encryption, Attribute-based encryption, Cloud security, Cloud computing

1. INTRODUCTION

Data outsourcing to the cloud are profitable for reasons of scalability,economy, and accessibility,but important technical challenges still remain. Many modifications to attribute-based encryption [1] are done to allow authorized users access to cloud data based on the required attributes so that the higher processing load from cryptographic processes is assigned to the cloud provider. Its cost-effective pay-per-use model generally results in a small part of the cost of deploying the computing resources in-house.

The amount of data is increasing exponentially by each passing minute; therefore an effective storage mechanism is needed so that the data redundancy is reduced. As the business organizations move more to the cloud environment the data on the cloud will keep on increasing hence duplication of data is certainly undesired. Another important requirement is for data to be accessible with fine-grained controls, to provide flexibility. A single user log-in is largely deficient in today’s data retrieval tasks.

The upcoming sections of this paper describe a new approach of key management comprising a data de-duplication mechanism. Section 3 describes the architecture of the proposed system and the methodology to implement it. Finally advantages of the system are stated along with the conclusion in last two sections.

2. LITERATURE SURVEY

Data exchange on the cloud has many solutions such that the cloud provider is not directly trusted.But some naïve solutions are difficult to scale at a large level. For example, the RSA algorithm [2] is dependent upon the factoring of large numbers. Theuser is given control over on the attribute level [7] but requires a trusted authority and restricts the owner with a pairing operation that is costly in the sense of computation.

Additionally, various proxy re-encryption schemes have been used for storage security. Onemethod is implemented by re-encrypting the stored content at the time of retrieval. Such technique can only be applied to an encrypted storage system in which the data owner implements a block encryption mechanism and the keys used in this mechanism are further encrypted to form a lockbox [9].

Proxy re-encryption is sometimes combined with CP-ABE [10] such that cloud provider forms the re-encryption keys based on a pre-shared secret between the data owner and the provider. Another related work suggests the combination of ABE and proxy re-encryption which allows fine-grained access control of resources while giving the responsibility of re-encryption to the cloud provider [11]. To avoid single point of failure, a multi-authority system has also been proposed [12]. Some approaches also require a trusted proxy for each decryption [13] but at the cost of increased communication overhead.

3. PROPOSED METHODOLOGY

A protocol of outsourcing data storage to a cloud in secure fashion is provided. The provider is inadequate to read saved data; authorized users may do so without arbitration by the data owner. An improvement is made over a traditional attribute based encryption model, so that responsibility over key generation is split between a data owner and a trusted authority; the user is freed of the highest computational burdens. Additional security is given through a group keying mechanism; the owner controls access based on the distribution of an additional secret key, beyond possession of the required attributes. Additional secret key is calculated by considering name of the file requested by the user, user's session ID, userID. This additional security measure is an optional variant applicable to sensitive data accessed frequently. Availability of data owner is a concern for the existing system because whenever a user requests for the data, owner has to grant the access explicitly. Due to this a lot of time is wasted and prolonged unavailability of owner will almost entirely block the access for the user.

To overcome this issue data owner's involvement is restricted for highly sensitive data only. User will be able to read all the data which is uploaded on the cloud server by the data owner but in order to download the data; owner's permission will be required. This mechanism helps in solving the problem caused due to the unavailability of the data owner for a large extent.

De-duplication helps to remove uploading of same files to the server; it automatically rejects all the similar files. It uses hashing algorithms to check for the replications of the files. It works on the block level i.e. it divides the contents of the files in blocks of data and then uploads or rejects the files depending upon the result of matching the output of hash function with hash functions of previous files. Figure 1 describes the working of hashing algorithm.

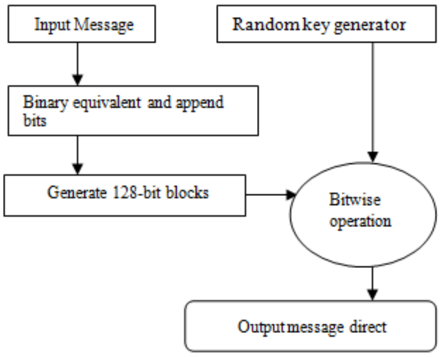

Figure 1: Working of hashing algorithm

Working of algorithm is divided into 6 steps:

Step 1:

Enter the input string.

Step 2:

Find the Similar binary string. (Make use of ASCII conversion for each character in the message string.)

Step 3:

Add a bit sequence ("01" here) to the string so that the length of the string is 64 bit shorter than a multiple of 512.

Append 64 more bits by scanning the string of step 3 (i) beginning from an arbitrary location (Here a rule can be implemented as the starting point as [length / 3])

Step 4:

Separate the output string of last step in 128 bit blocks.

Generate a 128 bit key using a random number generator.

Do a bitwise operation (like XOR, AND, OR, followed by Right Shift, Left Shift, zero fill shifting) among the 128 bit block and 128 bit random key.

Store the output of this step as stepwise message digest.

Step 5:

Do a bitwise operation between the current message digest and the previous message digest.

Jump to Step 4 until all the blocks of input message are finished.

Step 6:

Convert the output of Step 5 to its corresponding character value and save it as the final message digest.

The hashing function is an integral part of the system and all the outputs of the hash function are stored in a hash table for future use. Every time a new file is to be uploaded all the hash functions are matched with the hash function of the new file in order to check for duplication of data.

The working of the entire system is divided into 6 steps:

Step 1: In the first step, data owner uploads the data on the server and assigns read and download rights to the uploaded files. The data owner also assigns a secret key to each file which is used later as OTP for client's access.

Step 2: The de-duplication mechanism employed then converts the uploaded file into a hash function and matches it with previously generated hash functions in order to remove data redundancy.

Step 3: User has to first register on the system and then log into the system. At the time of registration user's credentials will be stored on the server for future use. User requests for the desired data and if it is not restricted for downloading, then the user can download it. If the requested files are restricted for downloading then data owner's permission will be needed in order to gain complete access. All the user requests have their own requested.

Step 4: For the restricted files data owner will receive the read or write requests from the user along with their respective request IDs and then the owner will grant read or download permission for the requested files. The requests can be viewed by the data owner and request contains request ID, name of the requestor, names of the files requested for. One data owner can view requests for the files which are self-uploaded and not by some other owner.

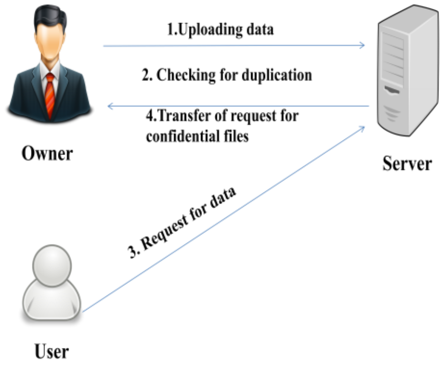
1.Uploading data
2. Checking for duplication
4.Transfer of request for confidential files
3. Request for data
Owner
Server
User

Figure 2: Working of system (steps 1-4)

Step 5: Once the data owner accepts the requests, all the users who requested for read and download access will obtain an OTP through E-mail.

Step 6: After entering the correct OTP user will have read or download access as per requested.

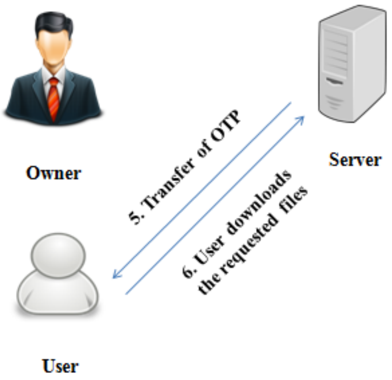

Figure 3: Working of system (steps 5-7)

4. ADVANTAGES

The proposed system gives following advantages over the existing systems:

1. System provides security to the confidential data of organizations and also provides better access control. Due to this unauthenticated users cannot gain access to confidential data. The data is stored on a cloud and only the users with current session key and valid credentials can get access to the data stored on the cloud. System also includes OTP mechanism. The OTP and valid key both together can only grant access to user

2. The de-duplication mechanism included in this proposed system allows efficient data storage on the cloud. This mechanism doesn't allowed uploading of multiple copies of same file

3. This mechanism reduces data owner's involvement in each and every download process on cloud. Hence the waiting time for client can be reduced. Also de-duplication mechanism allows for efficient storage of data on the cloud server.

4. This mechanism reduces data owner's involvement in each and every download process on cloud. Hence the waiting time for client can be reduced. Also de-duplication mechanism allows for efficient storage of data on the cloud server.

5. ACKNOWLEDGEMENTS

Wesincerelypaythankstoourguideandother staff members of the dept.ofInformationTechnology,AISSMSIOIT.

6. CONCLUSION

It is basically a key management system which increases the security of the cloud by providing better access control. Only authorized users are allowed to access the files on the cloud. Data owner's involvement in each and every data transfer is reduced but owner's control over important data is still intact. System can be used in business organizations, colleges, hospitals and all other places where important data is stored on cloud platform. Issue with the availability of data owner has been reduced by further refining owner's control over the data on the server.

De-duplication mechanism reduces the data redundancy on the server and hence provides efficient use of data storage. This mechanism works on block level and hence files with similar names but different content can also be uploaded on the server.

The system is scalable and can be used in almost any cloud environment with a power of growing in terms of security by various modern and better encryption techniques or multi-level authentication techniques as well.

7. REFRENCES

[1] P.K. Tysowski and M.A. Hasan, "Hybrid Attribute-Based Encryp-tion and Re-Encryption for Scalable Mobile Applications in Clouds," Technical Report 13, Centre for Applied Cryptographic Research (CACR), Univ. of Waterloo, 2013.

[2] R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,"Comm. ACM, vol. 26, no. 1, pp. 96-99, Jan. 1983.

[3] N. Balasubramanian, A. Balasubramanian, and A. Venkataramani, "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications," Proc. Ninth ACM SIGCOMM Conf. Internet Measurement Conf. (IMC '09),pp. 280-293, 2009.

[4] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption,"Proc. IEEE Symp. Security and Privacy (SP '07),pp. 321-334, 2007.

[5] Tassanaviboon and G. Gong, "OAuth and ABE Based Authorization in Semi-Trusted Cloud Computing: Aauth,"Proc. Second Int'l Workshop Data Intensive Computing in the Clouds (DataCloud-SC '11),pp. 41-50, 2011.

[6] X. Liang, R. Lu, and X. Lin, "Ciphertext Policy Attribute Based Encryption with Efficient Revocation," Technical Report BBCR, Univ. of Waterloo, 2011.

[7] J. Hur and D.K. Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems,"IEEE Trans. Parallel and Distributed Systems,vol. 22, no. 7, pp. 1214 1221, July 2011.

[8] Prof. RakeshMohanty, NiharjyotiSarangi, Sukant Kumar Bishi, "A Secured Cryptographic Hashing Algorithm" VSSUT, Burla, Orissa, India

[9] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with

Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, pp. 1-30, Feb. 2006

[10] Q. Liu, G. Wang, and J. Wu, "Clock-Based Proxy Re-Encryption Scheme in Unreliable Clouds," Proc. 41st Int'l Conf. Parallel Processing Workshops (ICPPW), pp. 304-305, Sept. 2012.

[11] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM '10, pp. 534-542, 2010.

[12] K. Yang and X. Jia, "Attributed-Based Access Control for MultiAuthority Systems in Cloud Storage," Proc. IEEE 32nd Int'l Conf. Distributed Computing Systems (ICDCS), pp. 536-545, 2012.

[13] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-Based Access Control in Social Networks with Efficient Revocation," Proc. Sixth ACM Symp. Information, Computer and Comm. Security (ASIACCS '11), pp. 411-415, 2011.