ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 6, June 2017

A Secure Deduplication Mechanism for Sensitive Data in Public Cloud Storage

[1] Deepa P.B, [2] Dr.Rudramurthy M.S
[1]Siddaganga Institute of Technology, Karnataka, [2] Siddaganga Institue of Technology, Karnataka
[1] geethrapriya.p.b@gmail.com,[2] rudrams2011@gmail.com

***Abstract*— — nowadays, cloud-based services are gaining more importance in leveraging services for large scale storage and distribution. These provide cost effectiveness, time saving and efficient utilization of computing resources. Many enterprises have decided to envision a staged migration for their data to public cloud, for better management of backup data. As data progressively grows, the cloud storage systems continuously face challenges in saving storage capacity. Data Deduplication gives solution to such challenges. It is a data compression technique for eliminating the redundant data. Assuming that cloud service provider may not be trustworthy (i.e. is honest but curious). The security, bandwidth and latency are of top concerns for an enterprise to store sensitive data in the public cloud storage. Considering these security challenges, we propose and implement a client side deduplication scheme for securely storing enterprise sensitive data via the pubic cloud. Efficiently and reliably managing the huge number of convergent keys by the user is a critical issue. Hence we have built a system where each convergent key is secured and the same is outsourced to the cloud. To avoid the weakness of convergent encryption, double encryption is done. By following this method, we can save storage space and cost. It also reduces bandwidth and ensures better confidentiality towards unauthorized users.**

***Index Terms*—Data deduplication, convergent encryption, and side channel attack.**

I. INTRODUCTION

Cloud Storage provider offers storage space for users, enterprises and organizations. Some of the commercial cloud storage services such as Microsoft Skydrive, Amazon S3 and Google Cloud Storage are used by millions of customers. There is much evidence as per studies shown in [1], by 2020 the analysis report states that the volume of data stored may cross 40 trillion gigabytes [3]. One of the major issues in cloud storage services is handling the increase in the amount of data. Recent studies by EMC[2] has revealed that today, 75% content are duplicated contents. To make data management scalable in cloud computing, cloud providers are constantly looking for techniques aimed to minimize redundant data. By taking the advantage of economic scale, the efficiency can be improved by using deduplication. Many cloud providers are adopting deduplication technique as it achieved high space and cost savings. Deduplication technique helps in removing the redundant data by storing only one copy of the data and providing link to other copies of the data. A number of deduplication schemes are proposed based on different deduplication strategies such as source or destination deduplication. Deduplication can be done at either the file-level or block level. For file-level deduplication, it removes the repeating files of contain same data. Deduplication can also be done at the block level, which removes identical blocks of data present in different files. The main advantage of deduplication is that it reduces storage capacity requirements and also provides benefits like lower power consumption, lower cooling, longer disk-based retention of data of data, and disaster recovery. Unfortunately, Deduplication and encryption are two conflicting technologies [4]. Although deduplication brings a lot of benefits, security and privacy issues will arise as user's sensitive data is vulnerable to both insider and outsider threats. Traditional encryption does not support the deduplication because distinct users encrypt their data with their key. This leads to different ciphertexts, making deduplication infeasible. Convergent encryption has been proposed to support the data confidentiality while making deduplication possible. It encrypts/decrypts a data with a convergent key, which is generated by computing the cryptographic hash value of the data. After key generation and data encryption, users maintain the keys and send the cipher text to the cloud. The cloud providers perform deduplication on the ciphertexts. The ciphertexts can only be decrypted by the identical user with their convergent keys. The convergent encryption is suitable but managing an enormous number of convergent keys is a critical task.

In this paper, "Secure deduplication mechanism for sensitive data in public clouds" is proposed. The proposed system consists of four entities. They are affiliate client, a Server, deduplication system and Public Clouds. For example, in an enterprise, in order to efficiently manage and save the cost of the data, it is moved to public clouds. Here a client is an employee who is allowed to store the data to the public cloud. Server is used for authentication of employee. The deduplication system identifies redundant file and does not allow same file to upload from a different user and also to provide security using convergent encryption at the client side and securely mange the key at the deduplication

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 6, June 2017

system. Convergent encryption suffers from the side channels attack for some extent. So here, the problem of convergent encryption is addressed by additional symmetric encryption before storing the data to public cloud.

II. LITERATURE SURVEY

The management of huge amount of archival storage systems consumes more power and equipments, as the scale increases [4]. Several deduplication methods have been attracted by the research community as the amount of data stored in the cloud is increasing gradually. The deduplication is used to minimize the storage space by the elimination of redundant data. The strategies can be classified based on fine grained data granularity, that is file level deduplication and block level deduplication. File level deduplication, which is a content aware that discovers the duplicate between files, if two files have the same hash value then they are redundant files. Removing the redundant files improve the storage capacity and also needs low computational overhead [7][6]. Block level deduplication is a removal of redundant blocks in the file and provides the index to that block, instead of storing the whole file. The file is divided into fine grained chunks using fixed sized blocks or variable sized blocks using Rabin fingerprinting to provide efficient deduplication ratio and then use hash value of each block to eliminate the block already stored in cloud.

In single user deduplication system, deduplication takes place only when the same user uploads same data. In cross user deduplication system, deduplication takes place only when the different user uploads same data, the data will not occupy a new storage space, and the service provider will create a reference of the original data for different users.

A. ***Server Side Deduplication***

Deduplication at the server side is performed after the upload of the file is done by the users. It's done in the background during the provisioning. The client does not know whether deduplication is done or not. So it is more secure than the client-side deduplication [8]

Harnik et al. [9] in their paper states many attacks such as Identifying Files, Learning the contents of Files and A covert channel. And they also analyze security issues and measures to reduce the risk of data leakage by using the deduplication in cloud storage. If an attacker gets the access to the storage can easily perform "dictionary attacks" by comparing the guessed data with the stored data and discover the predictable files. These attacks exploit the relationship between plaintext and the key in order to check if the copy of the file has already been stored or not. If attacker is the insider then he can also know who owns the file. For instance, if a user has stored illegal copies then it can be dangerous. Learn-the-remaining-information: attacker tries to guess the unknown information of the file by checking if the resulting file matches the stored data. For example, if user stores a pay slip, an attacker who knows the template may be able to get the salary details and other sensitive information by guessing missing information.

Bellare et al.[10] presented a scheme called DupLess, where the data owners of the duplicate data encrypts using the pseudorandom encryption key. So attacker cannot guess the plain data by using the hash value. With the help of independent key server, the brute-force attack can be reduced.

Liu et al.[8] exploited password authenticated key exchange protocol. User agrees for key agreement protocol with the previous users uploaded with same hash. After key is agreed using the protocol, user retrieves the key and encrypt the data. So ciphertexts is generated is same as the previously stored data. This avoids the side information leakage to unauthorized users and do not require additional servers. But it requires additional online user who has already stored the data and also may cause single point of failure. When compare to client-side deduplication, sever-side deduplication provides more security in terms of side information leakage. The goal of the cloud computing is to reduce the network bandwidth. So the trade-off between network bandwidth and data privacy can be seen.

B. *Client-side deduplication*

Deduplication is done before uploading the data. Here the user will compute hash value of the data and sends the tag to the cloud to check for duplication before outsourcing the data. Instead of uploading the content, only unique data is uploaded to the cloud storage. This saves the bandwidth. In traditional encryption algorithm, because the key is different, the same file will generate different ciphertexts. So the same file will be stored, which will seriously waste the storage space. In order to overcome this problem, Douceur has proposed a scheme called convergent encryption (CE), data owners encrypt the data with the convergent key K. K is computed from data content by using cryptographically strong hash function. This K is used as a key to the symmetric encryption. The data privacy is ensured by use of this Convergent encryption [4]. This will result in same ciphertext files for all same plaintext. The CE algorithm is described in three phases:

Key generation: user generates a convergent key k from the original data. Especially, a hash function $H1(F) = k$ is used as the generate the convergent key. $H1$ may be any strong cryptographic hash functions, like SHA-256.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 6, June 2017

Data encryption: To encrypt the content of the file or block and get the same cipher text C. Any symmetric key encryption algorithm can be used such that C = Encrypt(k, F), symmetric encryption algorithm like AES-256 can be used.

Data decryption: To get back the same content back. The convergent key k is used to decrypt the data as F = Decrypt(k,C). The decryption algorithm should be same as the encryption algorithm.

Warner and Pertula have proposed a scheme to overcome the dictionary attacks they have added a secret value to encrypt the data. Deduplication can be performed only by the users who share a secret value. Even though it overcome the weakness of the convergent encryption but limits the effectiveness of the deduplication. To perform deduplication, a tag is computed for the content and used to identify the duplicate at the cloud storage provider. Same data copies will generate same tags. And convergent key and their tags are always independently calculated. So from tags the convergent key should not be deduced and should not compromise the data confidentiality.

*Tag generation***:** File Tag generation algorithm uses another hash algorithm which will be saved at the storage provider with the ciphertexts C. $Tag(F) = H2(F)$, $H2$ is not same as the hash function used for the convergent encryption. Because of this property of the CE encryption algorithm many researchers have combined CE with different secure deduplication mechanisms in the multiuser environment.

Stanek et al[3] proposed in encryption technique that provide security for popular and unpopular data. For sensitive data, the conventional encryption is performed. And for the un sensitive data, two layered encryption is performed.

Bellare et al[8] used the third party server which identify the duplicates using file tags. And also reduces the bandwidth by not uploading the file. Key server will retain the keys that are generated by them. But if the attacker gains access to the secret, then the entire system is compromised and the confidently is not guaranteed.

C. *Key Management*

The convergent key management is done by encrypting the convergent key by using a master key of each user and stored securely at the user side. Later the file is stored corresponding to the encrypted data copies, in the public cloud storage. Dekey [16] Here the key is distributed across the multiple servers. It is still subject to brute-force attacks. Duan et al[17] specified a scheme, RSA-based threshold signature scheme is used to construct efficient, deterministic and non interactive scheme. Here the threshold signature is computed with the help of trusted dealer. But this concept is easily vulnerable for single point of failure.

III. SYSTEM MODEL

The proposed system, "Secure deduplication mechanism for sensitive data in public clouds", consists of four entities as shown in Figure 1.

- **Affiliated Clients**: The client encrypts the file using the convergent encryption key. The client secures the convergent key using xor operation. Then the client request for the file upload to the private cloud server. Only unique file is outsourced to the cloud and can be retrieve the data for later use.
- **Server**: It authenticates the clients and communicates between the clients and the deduplication system.
- **Deduplication system**: It is responsible for handling data deduplication, additional encryption and decryption of the file, storing metadata which includes keys. Hence avoiding the redundant copies of file by proving only the link to the user for the existing file
- **Public cloud**: public clouds are used for backup storage. They cannot be fully trusted so the double encrypted file is stored in the public cloud. This avoid the well know weakness of convergent encryption.

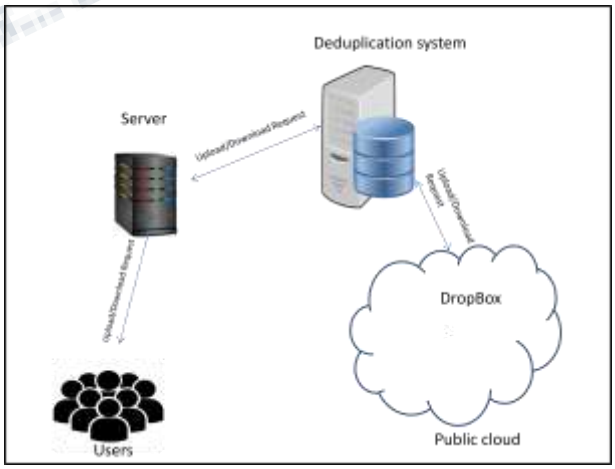

Fig 1: The system design of the proposed system

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 6, June 2017

IV. IMPLEMENTATION

Algorithm 1: Algorithm for upload of user File

Input: File and key from the user key u_k
Output: Upload data to Deduplication system
Begin:
C = FileEncrypt(File, u_k);
FileTag = Hash(C) ;
FileExist = checkReq(FileTag);
if *File Exists is false* **then**
UploadFile(File) ;
else
link of the file is provided to the user
end
End

Algorithm 2: Algorithm for encryption of File

Input: File to be encrypted, key from the user key u_k
Output: Encrypted File, s_k
Begin:
k = Hash(file) ;
C = AES(File, k) ;
s_k = k XOR u_k
return s_k,
End

Algorithm 3: Algorithm for DuplicationCheck(File, metadata)

Input: File, metadata ;
Output: Upload file to public cloud
Begin: FileTagExist = DuplicateCheck in the index table
if *FileTagExist is false* **then**
Store the file and its metadata in the Deduplication system
Return link of the file to user
Encrypt the file with the Deduplication server key and upload to public cloud
else
stores only the metadata information of the file in the Deduplication system
Return link of the file to user
end
End

We proposed a prototype for Secure deduplication mechanism. A Client module helps the clients to carry out the upload process. A server module is used to authenticate and for communication between clients and deduplication system. A Deduplication system module checks deduplication and stores the files. The public cloud which stores only the double encrypted files.

Upload phase: User who wants to upload file F to cloud, runs algorithm 1 after authenticating with server, user then encrypt the file F with the convergent key by using algorithm 2 and protects the convergent key k with user key u_k. The encrypted file C is denoted as C={c1,c2,c3,…..ct}. Then algorithm 1 calculates the hash of the encrypted file and sends request to the deduplication system to check the file exists or not with the metadata like user_id, key s_k, size. If the file does not exist then encrypted file should be uploaded. If the file exists then only link of the file is provided to the user. Deduplication system runs algorithm 3, checks the list of hash according to filetag(Hash of the value), and verifies if the file F has been stored. If the file F exists, it returns the reference to the user. If the file F does not exist, request the user to upload the file. Once a file is uploaded to the deduplication system then the file is encrypted with the deduplication system secret key and uploaded to the public cloud for future use.

Download Phase: when the user wants to access the stored file F, it sends the download request to the deduplication system. If the file is present in deduplication system, the download requests is sent to the public cloud for the file. The deduplication system decrypts the file with its secret key and sends the file to the user. After receiving the ciphertexts C, the user obtains server key s_k from the stored database and then XORed with the u_k to get the convergent key to decrypt the ciphertexts C.

V. CONCLUSION

In this paper, a secure deduplication technique is proposed. Deduplication is performed at the enterprise level, so it achieves cost space for the enterprise by saving the space in the cloud storage and upload bandwidth. And also provide better confidentiality for sensitive data by double encryption and decryption of the data and manages the key securely.

REFERENCES

[1] Yuan, Jiawei, and Shucheng Yu. "Secure and constant cost public cloud storage auditing with deduplication." *Communications and Network Security (CNS), 2013 IEEE Conference on*. IEEE, 2013.

[2] J. Gantz and D. Reinsel, "The digital universe decade – are you ready?"http://www.emc.com/collateral/analyst-reports/idc-digital-universe-are-you-ready.pdf, May 2010.

[3] Mi Wen, Kaoru Ota, He Li, Jingsheng Lei, Chunhua Gu, and Zhou Su. "Secure data deduplication with reliable key management for dynamic updates in cpss". *IEEETransactions on Computational Social Systems*, 2(4):137{147, 2015.

[4] John R Douceur, Atul Adya, William J Bolosky, P Simon, and Marvin Theimer. "Reclaiming space from duplicate _les in a serverless distributed _le system". *In Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*, pages 617{624. IEEE, 2002.

[5] I Nandhini, K Selvappriya, and S Tamilselvi. "A review of hilevel authorized deduplication method for cloud storage system". In *Advanced Computing and Communication Systems (ICACCS)*,

[6] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. "Message-locked encryption and secure deduplication". In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 296{312. Springer, 2013.

[7] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick PC Lee, and Wenjing Lou. "A hybrid cloud approach for secure authorized deduplication". *IEEE Transactions on Parallel and Distributed Systems*, 26(5):1206{1216, 2015.

[8] Jian Liu,Liu, N Asokan, and Benny Pinkas. "Secure deduplication of encrypted data without additional independent servers". In Proceedings of *the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 874{885. ACM, 2015.

[9] Danny Harnik, Benny Pinkas, and Alexandra Shulman-Peleg. "Side channels in cloud services: Deduplication in cloud storage". *IEEE Security & Privacy*, 8(6):40{47, 2010.

[10] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. "Dupless: Server-aided encryption for deduplicated storage". *IACR Cryptology* ePrint Archive, 2013:429.

[11] Jan Stanek, Alessandro Sorniotti, Elli Androulaki, and Lukas Kencl. "A secure data deduplication scheme for cloud storage". In *International Conference on Financial Cryptography and Data Security*, pages 99{118. Springer, 2014. 2016 3rd International Conference on, volume 1, pages 1{5. IEEE, 2016.

[12] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick PC Lee, and Wenjing Lou. "Secure deduplication with efficient and reliable convergent key management". *IEEE transactions on parallel and distributed systems*, 25(6):1615{1625, 2014.

[13] Yitao Duan. "Distributed key generation for encrypted deduplication: Achieving the strongest privacy". In Proceedings of *the 6th edition of the ACM Workshop on Cloud Computing* Security, pages 57{68. ACM, 2014.

[14] Pasquale Puzio, Re_k Molva, Melek Onen, and Sergio loureiro. "Cloudedup: secure deduplication with encrypted data for cloud storage". *In Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, volume 1, pages 363{370. IEEE, 2013.