IPHV8I3018X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

AN EFFICIENT DEDEDUPLICATION IN ENCRYPTED CLOUD DATA USING ATTRIBUTE-BASED STORAGE SYSTEM

[#1]AMEENA SAMREEN, M.Tech Student,
[#2]Dr.M.SUJATHA, Associate Professor
Department Of CSE,
JYOTHISHMATHI INSTITUTE OF TECHNOLOGICAL SCIENCES, KARIMNAGAR T.S.INDIA.

ABSTRACT: Cloud computing is a service by re-arranging resources in the Internet. Cloud service is popular for data storage. The data holder's privacy is the data stored in cloud and in the encrypted form. The cloud data deduplication is new challenges by the encrypted data, which is for processing in cloud and big data storage. The deduplication is not working on encrypted data. Data has wide applications in zones like keeping money, investigative exploration, prescription and government offices. Order is one of the ordinarily utilized assignments as a part of information mining applications. For back as decade, due to the ascent of different protection issues. The numerous hypothetical and commonsense answers for the order issue have been proposed under diverse security models. The late fame of distributed computing with notwithstanding, clients now has the chance to outsource their information, the information mining assignments and in encoded structure to the cloud. The information on the cloud is in existing security protecting characterization methods and encoded structures are not appropriate. In this paper, the characterization issue over encoded information in system concentrates on fathoming. System proposes a safe k-NN classifier over scrambled information in the cloud. The index is created with the help of Vector base cosine similarity (VCS) multiple strings matching algorithm which matches the pre-defined set of keywords with information in the data files to index them and store relevant data. The classification of information, the information access designs and convention ensures security of client's data inquiry is proposed in this system. To the best of their learning, there work is the first to add to a safe k-NN classifier over scrambled information under the semi-legitimate model.

Keywords: *Cloud computing, Access Control, Big Data, Data Deduplication, Encryption data, k-NN classifier, Security , outsourced databases.*

I.INTRODUCTION

With the potentially infinite storage space offered by cloud providers, users tend to use as much space as they can and vendors constantly look for techniques aimed to minimize redundant data and maximize space savings. A technique which has been widely adopted is cross-user deduplication. The simple idea behind deduplication is to store duplicate data (either files or blocks) only once. Therefore, if a user wants to upload a file (block) which is already stored, the cloud provider will add the user to the owner list of that file (block). Deduplication has proved to achieve high space and cost savings and many cloud storage providers are currently adopting it. Deduplication can reduce storage needs by up to 90-95% for backup applications [11] and up to 68% in standard file systems].

Along with low ownership costs and flexibility, users require the protection of their data and confidentiality guarantees through encryption. Unfortunately, deduplication and encryption are two conflicting technologies. While the aim of deduplication is to detect identical data segments and store them only once, the result of encryption is to make two identical data segments indistinguishable after being encrypted. This means that if data are encrypted by users in a standard way, the cloud storage provider cannot apply deduplication since two identical data segments will be different after encryption. On the other hand, if data are not encrypted by users, confidentiality cannot be guaranteed and data are not protected against curious cloud storage providers.

A technique which has been proposed to meet these two conflicting requirements is convergent whereby the encryption key is usually the result of the hash of the data segment. Although convergent encryption seems to be a good candidate to achieve confidentiality and deduplication at the same time, it unfortunately suffers from various well-known weaknesses including dictionary attacks: an attacker who is able to guess or predict a file can easily derive the potential encryption key and verify whether the file is already stored at the cloud storage provider or not.

Clouds are distributed computing systems built around core concepts such as computing as utility, virtualization of resources, on demand access to computing resources, and outsourcing computing services. Due to these concepts the clouds as an attractive platform for businesses enabling them to outsource some of their IT operations. Cloud computing system has an ability to provide on demand access to always-on computing utilities..cloud computing offers a new way to deliver services by rearranging resources over the

Internet and providing them to users on demand. It plays an important role in supporting data storage, processing, and management in the Internet of Things (IoT). Various cloud service providers (CSPs) offers huge volumes of storage to maintain and manage the IoT data, which can include videos, photos, and personal health records. These CSPs provide service properties, such as scalability, elasticity, fault tolerance, and pay per use. Thus, cloud computing has become a service paradigm to support IoT applications and IoT system deployment.

To ensure data privacy, existing research proposes to outsource only encrypted data to CSPs. However, the same or different users could save duplicated data by using different encryption schemes at the cloud. Although cloud storage space is big, duplication wastes networking resources, consumes excess power, and complicates data management. Thus, saving storage is becoming a crucial task for CSPs. Deduplication can achieve high space and cost savings. .At the same time, data owners want CSPs to protect their personal data from an unauthorized access. CSPs should therefore perform access control based on the data owners policies. the data owners want to control not only data access but also its storage and usage. The data deduplication should cooperate with data access control mechanisms. The same data, in an encrypted form, is only saved once at the cloud but can be accessed by different users based on the data owners policies. Data integrity can be provided to enhance the security of the data in the cloud. ECDSA, Elliptic Curve Digital Signature Algorithm can be used to check the integrity with the help of Trusted third party. After successful deduplication, data integrity can be checked to improve the security in the cloud.

In cipher text attribute base encryption scheme (CP-ABE) is a secure encryption technique use in cloud computing. In this scheme Data owner has full authority to assign all access permission .But In recent scenario data user are increase, so with the increasing number of cloud users there is a risk of users secret key will be escrow. Key of data owner will be manage or escrow because the key authority or cloud service provider both are not trusted. So to manage key of data owners and implement attribute with arbitrary state. So we propose a scheme with two party key issuing mechanisms with weighted attribute. Therefore both storage cost and encryption complexity for ciphertext are solve. The weighted attribute is introduced to not only extend attribute expression from binary to arbitrary state, but also to simplify access policy. Thus, the storage cost and encryption cost for a ciphertext can be relieved. We use the following example to further illustrate our approach. We propose an attribute-based data sharing scheme for cloud computing applications, which is denoted as ciphertext-policy weighted ABE scheme with removing escrow (CP-WABE-RE). It successfully resolves two types of problems: key escrow and arbitrary-sate attribute expression. We also provide deduplication check over files which are uploading on cloud by data owner to avoid duplicate file storage on cloud and also to save storage space require in cloud. A of ownership is provided to new user who uploads file on cloud and which is duplicate.

II. RELATED WORK

The current industrial deduplication solutions can't handle encrypted data. Existing solutions for deduplication are vulnerable to brute-force attacks and cant flexibly support data access control and revocation. It raises issues relating to security and ownership. Many users are likely to encrypt their data before outsourcing them to the cloud storage to preserve privacy, but this hampers de duplication because of the randomization property of encryption.

Message-Locked Encryption provides to achieve secure deduplication, goal currently targeted by numerous cloud storage providers. Message-Locke Encryption where the key under which encryption and decryption are performed is itself derived from the message. It is used in a wide variety of commercial and research storage service systems. A client first computes a key K H(M) by applying a cryptographic hash function H() to M, and then computes the ciphertext C E(K, M) via a deterministic symmetric encryption scheme. A second client encrypting the same file M will produce the same C, enabling deduplication.

Cloud storage has become a faster profit growth by a low cost, scalable, position for clients data. Since cloud computing environment is constructed based on architectures and interfaces, it has the ability to incorporate multiple internal and/or external cloud services together to provide high interoperability. The availability and integrity of outsourced data in cloud storages can be checked. There is a basic approach called PDP (Provable Data Possession). It is a proof technique for a storage provider to prove the integrity and ownership of clients data without downloading data. The proof-checking without downloading makes it especially important for large-size files and folders to check whether these data have been tampered with or deleted without downloading the latest version of data. PDP is used to ensure and enhance the integrity of data in the proposed system.

In cloud computing, data generated in electronic form are large in amount. To maintain this data efficiently, there is a necessity of data recovery services. There is a smart remote data backup algorithm, Seed Block Algorithm (SBA).The objective of this algorithm is twofold; first it help the users to collect information from any remote location in the absence of network connectivity and second to recover the files in case of the file deletion or if the cloud gets destroyed due to any reason. The time related issues are also being solved by SBA such that it will take minimum time for the

recovery process. SBA also focuses on the security concept for the back-up files stored at remote server, without using any of the existing encryption techniques. SBA can be used to efficiently recover lost data and also enhance data integrity.

The Elliptic Curve Digital Signature Algorithm is the Elliptic Curve analogue to the more widely used Digital Signature Algorithm (DSA).It is the application of ECC to digital signature generation and verification. Its security is based on the elliptic curve discrete logarithm problem. Elliptic Curve Digital signature represents one of the most widely used security technologies for ensuring un-forge-ability and nonrepudiation of digital data. The steps involved in ECDSA are formation of key-pair, signature-generation and signature-verification. The digital signature is typically created using the hash function. The transmitter sends the encrypted data along with signature to the receiver. The receiver in possession of senders' public key and domain parameters can authenticate the signature. ECDSA Provides more security. This algorithm is more secure against intruders. ECDSA is used to provide more security in the cloud environment.

III. CLOUDEDUP

The scheme proposed in this paper aims at deduplication at the level of blocks of encrypted files while coping with the inherent security exposures of convergent encryption.

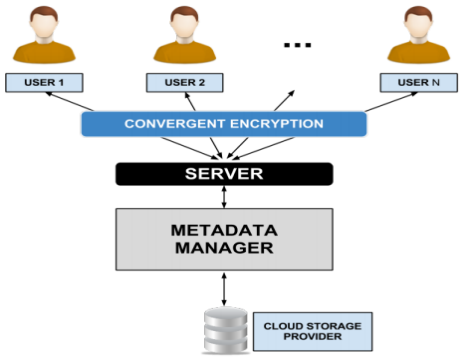

Fig. 1. High-level view of ClouDedup

The scheme consists of two basic components: a server that is in charge of access control and that achieves the main protection against COF and LRI attacks; another component, named as metadata manager (MM), is in charge of the actual deduplication and key management operations.

A. The Server

A simple solution to prevent the attacks against convergent encryption (CE) consists of encrypting the ciphertexts resulting from CE with another encryption algorithm using the same keying material for all input. This solution is compatible with the deduplication requirement since identical ciphertexts resulting from CE would yield identical outputs even after the additional encryption operation. Yet, this solution will not suffer anymore from the attacks targeting CE such as COF and LRI.

We suggest combining the access control function with the mechanism that achieves the protection against CE through an additional encryption operation. Indeed, access control is an inherent function of any storage system with reasonable security assurance. Enhancing the trusted component of the storage system, that implements access control, with the new mechanism against COF and LRI attacks, seems to be the most straightforward approach. The core component of ClouDedup is thus a server that implements the additional encryption operation to cope with the weaknesses of CE, together with a user authentication and an access control mechanism embedded in the data protection mechanism. Each data segment is thus encrypted by the server in addition to the convergent encryption operation performed by the user. As to the data access control, each encrypted data segment is linked with a signature generated by its owner and verified upon data retrieval requests. The server relies on the signature of each segment to properly identify the recipient.

B. Block-level

Deduplication and Key Management Even though the mechanisms of the server cope with the security weaknesses of CE, the requirement for deduplication at block-level further raises an issue with respect to key management. As an inherent feature of CE, the fact that encryption keys are derived from the data itself does not eliminate the need for the user to memorize the value of the key for each encrypted data segment. Unlike file-level deduplication, in case of block-level deduplication, the requirement to memorize and retrieve CE keys for each block in a secure way, calls for a fully-fledged key management solution. We thus suggest to include a new component, the metadata manager (MM), in the new ClouDedup system in order to implement the key management for each block together with the actual deduplication operation.

C. Threat Model

The goal of the system is to guarantee data confidentiality without losing the advantage of deduplication. Confidentiality must be guaranteed for all files, including the predictable ones. The security of the whole system should not rely on the security of a single component (single point of failure), and the security level should not collapse when a single component is compromised. We consider the server as a trusted component with respect to user authentication, access control and additional encryption. The server is not trusted with respect to the confidentiality of data stored at the cloud storage provider. Therefore, the server is not able to perform offline dictionary attacks. Anyone who has access to the storage is considered as a potential attacker, including employees at the cloud storage provider and the cloud storage provider itself. In our threat model, the cloud storage provider is honest but curious, meaning that it carries out its tasks but might attempt to

decrypt data stored by users. We do not take into account cloud storage providers that can choose to delete or modify files. Our scheme might be extended with additional features such as data integrity [16] and proofs of retrievability [20]. Among the potential threats, we identify also external attackers. An external attacker does not have access to the storage and operates outside the system. This type of attacker attempts to compromise the system by intercepting messages between different components or compromising a user's account. External attackers have a limited access to the system and can be effectively neutralized by putting in place strong authentication mechanisms and secure communication channels.

D. Security

In the proposed scheme, only one component, that is the server, is trusted with respect to a limited set of operations, therefore we call it semi-trusted. Once the server has applied the additional encryption, data are no longer vulnerable to CE weaknesses. Indeed, without possessing the keying material used for the additional encryption, no component can perform dictionary attacks on data stored at the cloud storage provider. The server is a simple semi-trusted component that is deployed on the user's premises and is in charge of performing user authentication, access control and additional symmetric encryption. The primary role of the server is to securely retain the secret key used for the additional encryption. In a real scenario, this goal can be effectively accomplished by using a hardware security module (HSM) [10]. When data are retrieved by a user, the server plays another important role. Before sending data to a given recipient, the server must verify if block signatures correspond to the public key of that recipient. The metadata manager (MM) and the cloud storage provider are not trusted with respect to data confidentiality; indeed, they are not able to decrypt data stored at the cloud storage provider. We do not take into account components that can spontaneously misbehave and do not accomplish the tasks they have been assigned.

IV. COMPONENTS

In this section we describe the role of each component.

A. User

The role of the user is limited to splitting files into blocks, encrypting them with the convergent encryption technique, signing the resulting encrypted blocks and creating the storage request. In addition, the user also encrypts each key derived from the corresponding block with the previous one and his secret key in order to outsource the keying material as well and thus only store the key derived from the first block and the file identifier. For each file, this key will be used to decrypt and re-build the file when it will be retrieved. Instead, the file identifier is necessary to univocally identify a file over the whole system. Finally, the user also signs each block with a special signature scheme. During the storage phase, the user computes the signature of the hash of the first block: $S0 = \sigma PKu\ (H(B0))$. In order not to apply costly signature operations for all blocks of the file, for all the following blocks, a hash is computed over the hash of the previous block and the block itself: $Si = H(Bi\ |Si-1)$. The main architecture is illustrated in Fig. 1.

B. Server

The server has three main roles: authenticating users during the storage/retrieval request, performing access control by verifying block signatures embedded in the data, encrypting/decrypting data travelling from users to the cloud and viceversa. The server takes care of adding an additional layer of encryption to the data (blocks, keys and signatures) uploaded by users. Before being forwarded to MM, data are further encrypted in order to prevent MM and any other component from performing dictionary attacks and exploiting the well-known weaknesses of convergent encryption. During file retrieval, blocks are decrypted and the server verifies the signature of each block with the user's public key. If the verification process fails, blocks are not delivered to the requesting user.

C. Metadata Manager (MM)

MM is the components responsible for storing metadata, which include encrypted keys and block signatures, and handling deduplication. Indeed, MM maintains a linked list and a small database in order to keep track of file ownerships file composition and avoid the storage of multiple copies of the same data segments. The tables used for this purpose are file, pointer and signature tables. The linked list is structured as follows:

- Each node in the linked list represents a data block. The identifier of each node is obtained by hashing the encrypted data block received from the server.
- If there is a link between two nodes X and Y, it means that X is the predecessor of Y in a given file. A link between two nodes X and Y corresponds to the file identifier and the encryption of the key to decrypt the data block Y.

The tables used by MM are structured as follows:

- **File table.** The file table contains the file id, file name, user id and the id of the first data block.
- **Pointer table.** The pointer table contains the block id and the id of the block stored at the cloud storage provider.
- **Signature table.** The signature table contains the block id, the file id and the signature.

In addition to the access control mechanism performed by the server, when users ask to retrieve a file, MM further checks if the requesting user is authorized to retrieve that file. This way, MM makes sure that the user is not trying to access someone else's data. This operation can be

considered as an additional access control mechanism, since an access control mechanism already takes place at the server. Another important role of MM is to communicate with cloud storage provider (SP) in order to actually store and retrieve the data blocks and get a pointer to the actual location of each data block.

D. Cloud Storage Provider (SP)

SP is the most simple component of the system. The only role of SP is to physically store data blocks. SP is not aware of the deduplication and ignores any existing relation between two or more blocks. Indeed, SP does not know which file(s) a block is part of or if two blocks are part of the same file. This means that even if SP is curious, it has no way to infer the original content of a data block to rebuild the files uploaded by the users. It is worth pointing out that any cloud storage provider would be able to operate as SP. Indeed, ClouDedup is completely transparent from SP's perspective, which does not collaborate with MM for deduplication. The only role of SP is to store data blocks coming from MM, which can be considered as files of small size. Therefore, it is possible to make use of well-known cloud storage providers such as Google Drive [7], Amazon S3 [3] and Dropbox [6].

E. A realistic example of ClouDedup

In this section we show that our proposed solution can be easily deployed with existing and widespread technologies. In the scenario we analyze, a group of users belonging to the same organization want to store their data, save as much storage space as possible and keep their data confidential. As shown in Fig. 2, the Server can be implemented using a Luna SA HSM [10] deployed on the users' premises.

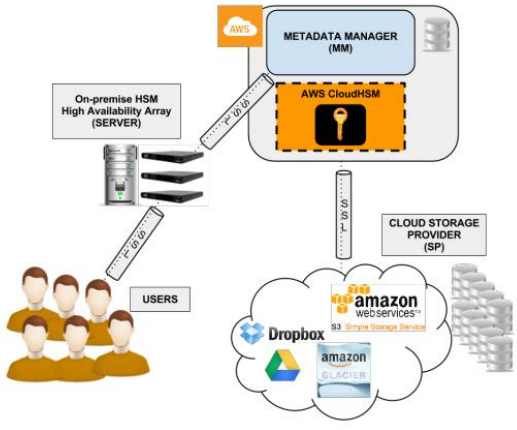

Fig. 2. A realistic example of ClouDedup

As documented in [8], in order to make the system resilient against single-point-of-failure issues, it is possible to build a high availability array by using multiple Luna SA HSMs. This way, in the case the main HSM crashes, it can be immediately replaced by an equivalent HSM without losing the secret key or getting worse performance. In order to guarantee data confidentiality even in the case the server is compromised, an additional HSM can be deployed between MM and SP. Deploying MM and the additional HSM in the same location, such as AWS [4], helps to minimize network latency and increase performance. This solution achieves higher security (it is very unlikely to compromise both HSMs at the same time) without significantly increasing the costs. MM can be hosted in a virtual machine on Amazon EC2 [1] and make use of a database to store metadata and encrypted keys. The additional HSM can be implemented by taking advantage of Amazon CloudHSM [5] which provides secure, durable, reliable, replicable and tamper-resistant key storage. Finally, very popular cloud storage solutions such as Dropbox [6], Amazon S3 [3], Amazon Glacier [2] and Google Drive [7] could be used as storage providers.

V. CONCLUSION

We designed a system which achieves confidentiality and enables block-level deduplication at the same time. Our system is built on top of convergent encryption. We showed that it is worth performing block-level deduplication instead of filelevel deduplication since the gains in terms of storage space are not affected by the overhead of metadata management, which is minimal. Additional layers of encryption are added by the server and the optional HSM. Thanks to the features of these components, secret keys can be generated in a hardwaredependent way by the device itself and do not need to be shared with anyone else. As the additional encryption is symmetric, the impact on performance is negligible. We also showed that our design, in which no component is completely trusted, prevents any single component from compromising the security of the whole system. Our solution also prevents curious cloud storage providers from inferring the original content of stored data by observing access patterns or accessing metadata. Furthermore, we showed that our solution can be easily implemented with existing and widespread technologies. Finally, our solution is fully compatible with standard storage APIs and transparent for the cloud storage provider, which does not have to be aware of the running deduplication system. Therefore, any potentially untrusted cloud storage provider such as Amazon, Dropbox and Google Drive, can play the role of storage provider.

REFERENCES

[1] A. Balu and K. Kuppusamy. "An expressive and provably secure ciphertext-policy attribute-based encryption."Information Sciences, 276(4):354– 362, 2014.

[2] M. Chase and S. S. Chow. "Improving privacy and security in multiauthority attribute-based encryption."Proceedings of the 16th ACM Conference on Computer and Communications Security, pages 121–130,2009.

[3] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. "Randomizable proofs and delegatable anonymous credentials". Proceedings of the

29th Annual International Cryptology Conference, pages 108–125, 2009.
[4]Junbeom Hur, Dongyoung Koo, "Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage,"IEEE Transactions on Knowledge and Data Engineering DOI 10.1109/TKDE.2016.2580139,
[5] J. Bethencourt, A. Sahai, and B. "Waters. Ciphertext-policy attribute based encryption,"IEEE Symposium on Security and Privacy, pages 321– 334, 2007.
[6] C. Fan, S. Huang, and H. Rung. Arbitrary-state attribute-based encryption with dynamic membership. IEEE Transactions on Computers, 63(8):1951–1961, 2014.
[7] Vipul Goyal,Omkant Pandey, Amit Sahai, Brent Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," ACM 1-59593-518-5/06/0010
[8] T. Jiang, X. Chen, J. Li, D. S. Wong, J. Ma, and J. K. Liu. Towards secure and reliable cloud storage against data re-outsourcing. Future Generation Computer Systems, 52:86–94, 2015.
[9] Yan Zhu, Hongxin Hu, Gail-Joon Ahn, and Mengyang Yu. Cooperative provable data possession for integrity verification in multicloud storage. IEEE transactions on parallel and distributed systems, 23(12):2231– 2244, 2012.
[10] Kruti Sharma and Kavita R Singh. Seed block algorithm: A remote smart data back-up technique for cloud computing. In Communication Systems and Network Technologies (CSNT), 2013 International Conference on, pages 376– 380. IEEE, 2013
[11] Opendedup. http://opendedup.org/.
[12] Atul Adya, William J Bolosky, Miguel Castro, Gerald Cermak, Ronnie Chaiken, John R Douceur, Jon Howell, Jacob R Lorch, Marvin Theimer, and Roger P Wattenhofer. Farsite: Federated, available, and reliable storage for an incompletely trusted environment. ACM SIGOPS Operating Systems Review, 36(SI):1–14, 2002.
[13] Mihir Bellare, Alexandra Boldyreva, and Adam ONeill. Deterministic and efficiently searchable encryption. In Advances in Cryptology-CRYPTO 2007, pages 535–552. Springer, 2007.
[14] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. Dupless: Server-aided encryption for deduplicated storage. 2013.
[15] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. Message-locked encryption and secure deduplication. In Advances in Cryptology–EUROCRYPT 2013, pages 296–312. Springer, 2013.