DEDUPLICATION OF DATA IN COMMUNITY CLOUD

[1]I.Mettildha Mary, [2]S.Aishwaryaa, [3]A.Maria Vinolia, [4]Jothisa Prabhakar
[1]Assistant professor (Sr.Gr), Department of Information Technology,
Sri Ramakrishna Engineering College,
[2,3,4]UG Scholar, Department of Information Technology,
Sri Ramakrishna Engineering College.

Abstract

Cloud computing offers a new way of service provision by re-arranging various resources over the Internet. The most important and popular cloud service is data storage. Deduplication, which can save storage cost by enabling us to store only one copy of identical data, becomes unprecedentedly significant with the dramatic increase in data stored in the community cloud. For the purpose of ensuring data confidentiality, they are usually encrypted before outsourced. It integrates cloud data deduplication with access control.

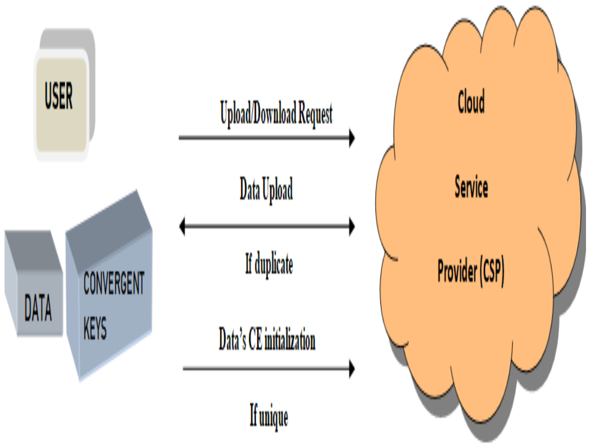
Keywords: Deduplication, Convergent Encryption(CE), Advanced Encryption Standards (AES), Cloud Service Provider(CSP).

I. Introduction

Cloud computing are shared pools of configurable computing system resources and higher-level services which will be speedily provisioned with borderline management effort, often over the Internet. Cloud computing depends on sharing of resources to realize coherence and economies of scale, similar to a public utility. It's known as cloud computing as a result of the knowledge being accessed is found in "the cloud" and doesn't need a user to be during a specific place to achieve access to it internal failure, and pay-per-utilize. The most essential and well known cloud administration is information stockpiling administration. Cloud clients transfer individual or classified information to the server farm of a Cloud Service Provider (CSP) and permit it to keep up these information. Since interruptions and assaults towards touchy information at CSP are not avoidable, it is reasonable to expect that CSP can't be completely trusted by cloud clients. Also, the loss of control over their very own information prompts to big data security dangers, particularly information protection spillages. Because of the quick improvement of information mining and different examination innovations, the security issue gets to be distinctly genuine. Subsequently, a great practice is to just outsource encoded information to the cloud with a specific end goal to guarantee information security and client protection. Be that as it may, the same or diverse clients may transfer copied information in scrambled frame to CSP, particularly for situations where information is shared among numerous clients. Despite the fact that Cloud storage space is immense, information duplication extraordinarily squanders organize assets, devours a great deal of vitality, and entangles information administration. The advancement of various administrations additionally makes it pressing to convey productive asset administration instruments. Thus,deduplication gets to be distinctly basic for big data stockpiling and handling in the cloud.

II. System Design Architecture

It consists of two modules such as user and admin. In cloud it is referred to as data owner and data holder. The required data to be uploaded in the cloud is attached with convergent keys and are then uploaded as an encrypted data into the CSP. It return if there exists a duplicated data.



III. Existing System

Most existing solutions cannot ensure reliability; security and privacy with sound performance Furthermore, to support deduplication, a short cryptographic hash value of the content will also be computed and sent to each storage server as the fingerprint of the fragment stored at each server. Only the data owner who first uploads the data is required to compute and distribute such secret shares, while all following users who own the same data copy do not need to compute and store these shares any more. To recover knowledge copies, users must access a minimum number of storage servers through authentication and obtain the secret shares to reconstruct the data. In different words, the secret shares of data will only be accessible by the authorized users who own the corresponding data copy. Another distinguishing feature of our proposal is that data integrity, including tag consistency, can be achieved. Existing solutions for deduplication suffer from brute-force attacks. They cannot flexibly support data access control and revocation at the same time. Most existing solutions cannot guarantee responsibleness, security and privacy with sound performance.

IV. Proposed System

The New deduplication scheme KeyDup without such a strong assumption. The 3 main contributions of this paper square measure summarized as follows.

- We propose a novel client-side deduplication scheme. Specifically, we make a combination of convergent encryption (CE) and Advanced Encryption Standards(AES) to achieve secure and efficient space management in community cloud.
- Security analysis demonstrates that our scheme ensures the confidentiality of data files and the security of convergent keys.
- A comprehensive performance comparison between KeyDup and several present works is given, showing that our scheme makes a better tradeoff among the storage cost, communication overhead and computation overhead.

V. Convergent Encryption

Convergent encryption, introduced by Douceur et.al. has been widely used in the deduplication of data stored in the cloud. It is a cryptosystem that produces identical cipher text files from identical plaintext files, irrespective of their encryption keys. To encrypt a data copy (a file or a block) using convergent encryption, a user first computes a cryptographically strong hash value from the data copy, and then using this hash value as the convergent key to encrypt the data copy. The user could also derive a tag for the data copy, which will be used to detect duplication. In this way, the same data copy will be encrypted by the same key, resulting in the same cipher text and tag. Then the cipher text and the tag are given to the server and the user retains the convergent key. The server can now perform deduplication on the cipher text, checking whether or not itis already stored. Note that both the convergent key and the tag are independently derived and the tag cannot be used to deduce the convergent key and compromise data confidentiality. A convergent encryption scheme can be formally defined as a tetrad of the following four algorithms (KeyGen, Encrypt, Decrypt, TagGen) :

- KeyGen(M) → K is the key generation algorithm that maps a data copy M to a convergent key K;
- Encrypt(K,M) → C is the symmetric encryption algorithm that takes both the convergent key Kand the data copy M as inputs and then outputs a ciphertext C;
- Decrypt(K,C) → M is the decryption algorithm that takes both the ciphertext C and the convergent key K as inputs and then outputs the original data copy M;
- T agGen(M) → T (M) is the tag generation algorithm that maps the original data copy M and outputs a tag T (M).

V. Proof of Ownership

Deduplication requires that if two or more users own the same file, only a single copy should be stored in the community cloud. When users upload a file that has been existing, he will prove his ownership of the file to the user and obtains the information associated with the storage, such as the pointer of the copy. The user directly sending a hash value of the data copy to the server for verification seems like a possible solution. However, as the data is encrypted, the cloud server cannot compute the hash value of the data copy and needs to store many hash values along with data copies, especially when the amount of data is huge first proposed the notion of proof of ownership (PoW), which enables the client to prove to the server that he has a copy of the file, without actually sending the file. It's essentially an interactive protocol performed between a prover (user) and a verifier (server). The verifier first derives a short value(M) from the data copy M.

VI. Advanced Encryption Standards

AES uses the same secret key is used for the both encryption and decryption. Unlike AES 128 bit encryption and decryption, if we need a stronger AES 256 bit key, we need to have Java cryptography extension (JCE. We adopt the Advanced Encryption Standards to encrypt convergent keys before sending them to the Cloud Service Provider (CSP). A necessary authority involved in an AES is the Private Key Generator PKG. Using its master secret key MSK; the PKG can generate a decryption key for each new member with identity ID to decrypt messages. An attractive feature of the AES scheme is that the data holder does not hold any private information. Messages can be encrypted with the help of a public key and the set of identities of the receivers. Then all the identities in S are able to decrypt the messages. Given security parameterS(k) and maximal size m of the target set, an AES scheme can be formally described as a tuple of algorithms (Setup, Extract, Encrypt, Decrypt):Setup(M). Takes as input the security parameter s(k) and m the maximal size of the set of receivers for one encryption, and outputs a master secret key MSK and a public key PK. Extract(MSK, ID). Takes as input the master secretkey MSK and a user identity ID. Extract generates a user private key Encrypt(S, PK). Takes as input the public key PK and a set of included identities S = {ID1,IDs} with s ≤ m, and outputs a pair (Hdr,K). When a message M{0, 1} is to be broadcast to users in S, the broadcaster generates (Hdr,M) ← Encrypt(S, PK), computes the encryption CM of M under the symmetric key Mand encrypt (Hdr, S,CM). We will refer to Hdr as the header (which actually encapsulates the advanced encryption) of broadcast ciphertext, K as the message encryption key and CM as the broadcast body.

VII. Literature Survey

7.1 RECLAIMING SPACE FROM DUPLICATE FILES IN A SERVERLESS DISTRIBUTED FILE

The Farsite distributed file system provides availability by replicating each file onto multiple desktop computers. Since this replication consumes significant storage space, it is important to reclaim used space where possible. Measurement of over 500 desktop file systems shows that nearly half of all consumed space is occupied by duplicate files. We present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Our mechanism includes 1) convergent encryption, which enables duplicate files to coalesced into the space of a single file, even if the files are encrypted with different users' keys, and 2) SALAD, a Self- Arranging, Lossy, Associative Database for aggregating file content and location information in a decentralized,scalable, fault-tolerant manner. Large-scale simulation experiments show that the duplicate-file coalescing system is scalable, highly effective, and fault-tolerant.

7.2 SECURE DEDUPLICATION WITH EFFICIENT AND RELIABLE CONVERGENT KEY MANAGEMENT

Data deduplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. Promising as it is, an arising challenge is to perform secure deduplication in cloud storage. Although convergent encryption has been extensively adopted for secure deduplication, a critical issue of making convergent encryption practical is to efficiently and reliably manage a huge number of convergent keys. This paper makes the first attempt to formally address the problem of achieving efficient and reliable key management in secure deduplication. We first introduce a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates an enormous number of keys with the increasing number of users and requires users to dedicatedly protect the master keys. To this end, we propose Dekey , a new construction in which users do not need to manage any keys on their own but instead securely distribute the convergent key shares across multiple servers. Security analysis demonstrates that Dekey is secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs limited overhead in realistic environments.

7.3 MULTIPLE RAMP SCHEMES

A (t,k,n,S) ramp scheme is a protocol to distribute a secret s chosen in S among a set P of n participants in such a way that:

(1) sets of participants of cardinality greater than or equal to k can reconstruct the secret.

(2) sets of participants of cardinality less than or equal to t have no information on s, whereas

(3) sets of participants of cardinality greater than t and less than k might have "some" information on s.

In this correspondence we analyze multiple ramp schemes, which are protocols to share many secrets among a set P of participants, using different ramp schemes. In particular, we prove a tight lower bound on the size of the shares held by each participant and on the dealer's randomness in multiple ramp schemes.

VIII.SECURITY ANALYSIS AND PERFORMANCE EVALUATION

8.1 Security Analysis

Our scheme provides a secure approach to protect and deduplicate the data stored in cloud by concealing plaintext from both CSP and AP. The security of the proposed

Scheme is ensured by CE theory, symmetric key encryption and hashing theory.

Proposition 1. The cooperation of CSP and AP without collusion guarantees that only eligible users can access plain data M and the data can be deduplicated in a secure way.

Proof. CSP has no way to know M since it is always in an encrypted form. CSP knows CK encrypted with pkAP , but AP does not share its own secret key skAP with CSP.

Thus CSP cannot know DEK and then M. AP has no way to access M since its access is blocked by CSP although AP could obtain DEK. In addition, we apply proper management protocols to support data storage management and data owner management to achieve deduplication at

the same time.

8.2 Convergebt Encryption Time

In the implementation, we used AES for symmetric encryption and tested three different sized AES keys: 128-bit, 192-bit and 256-bit. Figure 6 shows the CP-ABE encryption and decryption time of each sized AES key, regarding to different access policies (i.e., authorized individual CEs). We find that the AES key size has no much effect on the performance of convergent encryption and decryption. For different individual CE's requested for data access, the encryption time varies because different numbers of authorized CE's are enabled in the access policy. The higher the required CE is, the less time the encryption process spends.

IX. CONCLUSION AND FUTURE SCOPE

In this paper, we propose a secure client-side deduplication scheme KeyDup to effectively manage convergent keys. Data deduplication in our design is achieved by interactions between data owners and the Cloud Service Provider (CSP), without participation of other trusted third parties or Key Management Cloud Service Providers. The security analysis shows that our KeyDup ensures the confidentiality of data and security of convergent keys, and well protects the user ownership privacy at the same time. Experimental results demonstrate that the security of our scheme is not at the expense of the performance. For our future work, we will try to seek ways to protect the identity privacy of data owners, which is not considered in our scheme and also the process of splitting the content into block level data which provides accuracy of deduplication will be the future scope.

REFERENCES

[1] Amazon Web Services.Available: https://aws.amazon.com/cn/.

[2] D.A. Sarma, X. Dong, and A. Halevy, Bootstrapping pay-as-you-go data integration systems[C]. ACM SIGMOD International Conferenceon Management of Data, SIGMOD 2008, Vancouver, Bc, Canada,June. DBLP, 2008:861-874.

[3] J.R. Douceur, A. Adya, W.J. Bolosky, D. Simon, and M. Theimer,Reclaiming Space from Duplicate Files in a Server less DistributedFile System[C]. Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on. IEEE, 2002: 617-624.

[4] S. Ghemawat, H. Gobioff, and S. Leung, The Google File System[M].SOSP ’03 Proceedings of the nineteenth ACM symposium on Operating systems principles, 2003, 37(5): 29-43.

[5] D. Borthakur, HDFS architecture guide[J]. Hadoop Apache Project,2008, 53.

[6] J. Li, X. Chen, M. Li, J. Li, P.P.C. Lee, and W. Lou, Secure Deduplication with Efficient and Reliable Convergent Key Management[J]. IEEEtransactions on parallel and distributed systems, 2014, 25(6): 1615-1625.

[7] G.R. Blakley and C.A. Meadows, Security of Ramp Schemes[C].Crypto. 1984, 84: 242-268.

[8] A.D. Santis and B. Masucci, Multiple Ramp Schemes[J]. IEEE Transactions on Information Theory, 1999, 45(5): 1720-1728.

[9] M. Wen, K. Ota, H. Li, J. Lei, C. Gu, and Z. Su, Secure Data Deduplication with Reliable Key Management for Dynamic Updates in Cpss[J]. IEEE transactions on computational social systems, 2015,2(4): 137-147.

[10] W. Leesakul, P. Townend, and J. Xu, Dynamic Data Deduplication in Cloud Storage[C]. IEEE, International Symposium on ServiceOriented System Engineering. IEEE, 2014:320-325.