ISSN: 0976-3104

SPECIAL ISSUE: (Emerging Technologies in Networking and Security (ETNS)

Keerthana et al.

ARTICLE OPEN ACCESS

A SURVEY ON MANAGING CLOUD STORAGE USING SECURE DEDUPLICATION

K . Keerthana*, C. Suresh Gnanadhas, RT. Dinesh Kumar

Dept of Computer Science and Engineering, Vivekanandha College of Engineering for Women, Namakkal, T.N., INDIA

ABSTRACT

Aim: *Cloud is used as a storage platform to store various types of data. Many commercial and sensitive data are available in file format. So, we focus on storing data files in cloud without any replicates. Removing replicates will help the cloud to handle its data efficiently. The process of removing replicates is known as deduplication. But while storing, the files undergo client side encryption to elevate the privacy of user files. This challenges both the user and server to find the duplicate files over encrypted data. Many recent deduplication schemes have been proposed to overcome this challenge. But during client side encryption, all the schemes encrypt the files using symmetric encryption algorithm. In this paper, we survey about various recent deduplication schemes which mainly focus on file – level deduplication and here we analyze the efficiency of existing schemes. Based on the analysis we propose attribute and policy based dedupe system to enhance the security of deduplication process that has efficiency higher than the existing solutions.*

Published on: 2nd -December-2016

KEY WORDS

Cloud, Deduplication, Client side encryption, File – level deduplication, Proof-of-ownership

***Corresponding author: Email:** k.keerthanakarthikeyan@gmail.com; **Tel.:** +91 9965727694

INTRODUCTION

In this Internet era data grows rapidly. Storing huge amount of data locally using Pcs, Pen drive, CD (or) DVD is highly impossible. So the users tend to use the cloud for storage purpose. Cloud is a virtual environment where the user can use remote servers through internet for storing, managing or for processing the data. Cloud storage is one of the services offered by cloud. It provides high security of data, reduce the cost of storage, easy sharing of huge data, data recovery etc. So, using the cloud storage space efficiently is essential for every user. For that purpose data deduplication technique is used.

OBJECTIVE

Data Deduplication is a method of data compression that eliminates the replicates of data. This process will only save the unique copy of data in a storage media. When a data redundancy occurs it provides a pointer to access that data and ignore the process of saving the redundant data in storage. This process automatically saves the storage space. Reducing the storage space requirements will reduce the cost on disk expenditures, bandwidth usage for transacting data over internet and helps us to use the storage space efficiently. It identifies the redundant data by comparing the data already present in the storage. For preserving privacy of data the user may encrypt a data before uploading it in cloud storage. Then comparing an encrypted data to recognize the redundant data is a challenging task. To overcome this challenge we use convergent encryption algorithm [15]. This algorithm will produce identical cipher text for identical plain text. This algorithm is also known as Content Hash Keying.

SCOPE

The main aim of this paper is to focuses on File level deduplication which is one type of deduplication. Some types of deduplication which is used in our existing schemes are explained below

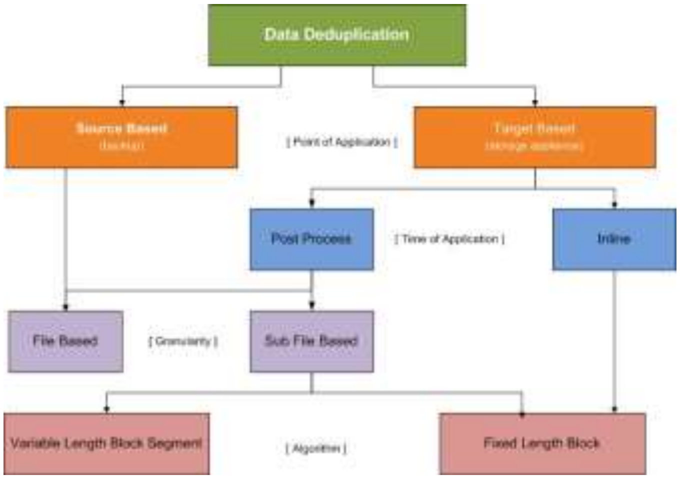


Fig. 1:Types of deduplication

……

Source based deduplication is also known as client-side deduplication. This process removes the redundant data before transmitting the data to storage (i.e., target). It efficiently decreases the bandwidth usage and storage space where as Target Based deduplication is also known as server-side deduplication. Here the process of removing redundant data takes place within the target system. This method is independent of hardware and software used in client side. Deduplication takes place even when the user is offline.

Post process deduplication is an asynchronous process [11] of removing redundant data after the data is written to the storage and Inline deduplication is a process of removing redundant data before it is written to the storage.

File based deduplication is a process of removing redundant files. It involves calculating single hash value for a file and that hash value is compared against already stored hash value present in cloud storage. If it matches then the file is already present in the storage and then only the pointer will be saved in the storage. The other type is Sub File based deduplication. It is a process of removing redundant blocks present in the files. It split the file into variable size block or fixed size block and calculates the hash value for each block to find the replicates present in the storage.

ANALYSIS OF EXISTING SYSTEM

This section presents the study on Existing deduplication schemes. This study helps us to find the drawbacks present it in the existing system. Based on this analysis, the proposed system will be designed.

Table 1: COMPARITIVE STUDY ON DEDUPLICATION SCHEMES

SI.NO	TITLE	Type of Deduplication	Algorithms Used	Merits	Demerits
1.	Deduplication on Encrypted Big Data in Cloud	File Level Deduplication	Deduplication Algorithm: PRE (proxy re-encryption) Encryption algorithm: AES Hashing Algorithm: SHA 1 Data Ownership Verification: ECC (Elliptic curve cryptography)	Resist offline brute force attack, low cost and access control over encrypted data	Hash function and symmetric encryptions are not highly secured.

2.	Encrypted Data Management with Deduplication in Cloud Computing	File Level Deduplication	Deduplication Algorithm: CP-ABE Encryption algorithm: AES Hashing Algorithm: SHA 1 Data Ownership Verification: RSA	Low operational and implementation cost. Doesn't depend on third party	Consume more time for key generation.
3.	A Hybrid Cloud Approach for Secure Authorized Deduplication	File Level Deduplication	Encryption algorithm: AES Tag Generation: SHA-1 Token Generation: HMAC-SHA-1	Resilient to Insider and outsider attack.	Vulnerable to Brute force attack.
4.	A Scheme to Manage Encrypted Data Storage with Deduplication in Cloud		Deduplication Algorithm: PRE Encryption algorithm: AES Data Ownership Verification: RSA	Resist offline brute force and dictionary attack	Cannot implemented directly in cloud
5.	Secure Deduplication with Dynamic Ownership Management in Cloud Storage	File Level deduplication and Server side deduplication	Encryption algorithm: AES with electronic code book algorithm Key and token generation: MD5 Deduplication Algorithm : Randomized convergent Encryption	Secured against chosen-plaintext attack, collusion attack and poison attack. It has forward and backward secrecy.	Exposed to data loss attack.
6.	Encrypted Deduplication in Cloud Storage	Client side **deduplication /** Server side deduplication	Hash Function : SHA 2 Encryption algorithm: AES,RSA(Asymmetric encryption) and Elgamal (homomorphic encryption)	Secured cipher text.	High Cost
7.	HEDup: Secure Deduplication with Homomorphic Encryption	File level deduplication	Encryption algorithm: AES Hash Function : MD5 Or SHA	Resilient to Man in middle attack	Resilient to Passive attacks .

8.	BDO-SD: An Efficient Scheme for Big Data Outsourcing with Secure Deduplication	Support both File level deduplication and block level deduplication	Encryption algorithm: AES Hash Function : SHA 1 Data Ownership Verification: blind signature system	Resist Brute force attack	communication overhead
9.	Secure Deduplication of Encrypted Data without Additional Independent Servers	File Level deduplication	Deduplication algorithm: PAKE Key generation: AES Encryption algorithm: Elgamal Hashing Function: SHA 256	Secure with Online brute force attack.	Exposed to Offline brute force attack, dictionary attack, week passwords and. Block level deduplication will cause additional overhead.
10.	Enhanced Secure Thresholded Data Deduplication Scheme for Cloud Storage	File Level deduplication	Encryption algorithm: AES – 128 – CTR	Resist User collusion attack & Low cost	Vulnerable to attacks based on knowledge of the hash of the plaintext file

PROPOSED SYSTEM

This section contains a description about the proposed work namely Attribute and policy based Dedupe System. It adds security to the deduplication process by using various factors such as certificates, signature, access control policies, hashing algorithm etc.

Attribute and policy based Dedupe System

In this system, we introduce access control policies to secure the user data. Two types of access control policies are used. One is providing read only access to user and the other is providing read & write access to the user. These policies will be provided based on the attributes of the user. Data owner and Data holder are the two kinds of user attributes. For data owner both read & write access will be provided and for data holders only read access will be provided.

Terms involved in Attribute and policy based Dedupe System

Key generation

The user should generate key based on their attribute and access control policy system [APS]. For E.g., if U1 is a data Owner who has Read Write access then his key is ORW23 where 23 is a random number chosen by user. This key is used as public key of user [PK]. Then the user should generate key pair for hashing algorithm (SHA 2). The dedupe checker will generate its key pair using CP ABE [cipher policy attribute based encryption]. And public key of dedupe checker is distributed to all CSP Users

Token and Signature generation

Using SHA 2 algorithm the user can generate data token [hash value of their file]. Then the signature is generated by encrypting the hash value using public key of a user.

Re encryption Key

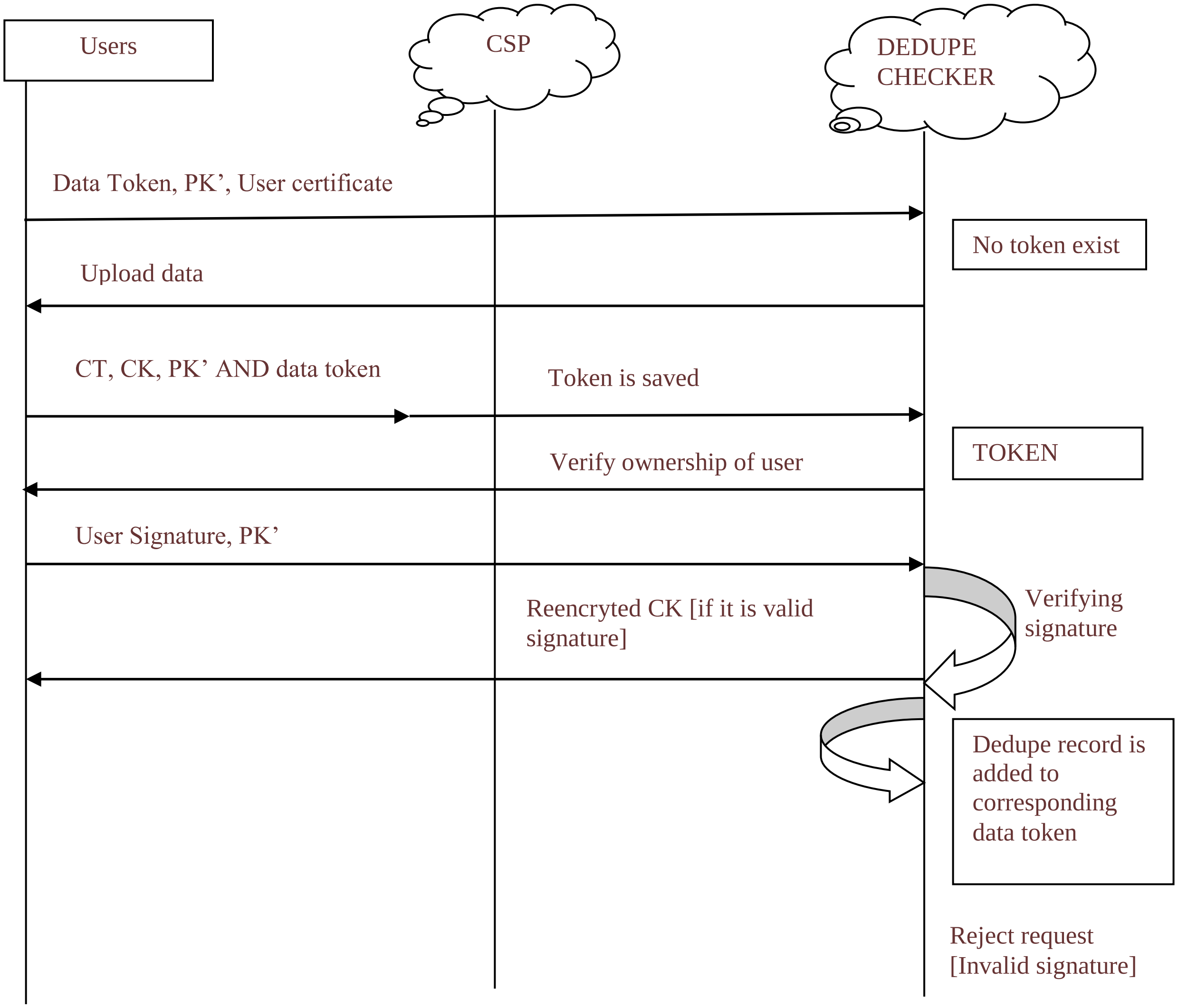
If dedupe occurs and the user is a valid data holder then dedupe checker will generate the reencryption key. The corresponding CK of that data which is stored in CSP is reencrypted by the public key of the data holder who proves their ownership.

Signature Validation

When dedupe checker receives signature and encrypted public key PK’ of user for verification. First the dedupe checker will decrypt the public key using its secret key. Then the signature is decrypted by obtained user public key. Finally a hash value will be produced. If that hash value matches with the data owner hash value then he/she is a valid data holder.

SYSTEM ARCHITECTURE

The system architecture of Attribute and policy based Dedupe System is given below


Users
CSP
DEDUPE CHECKER
Data Token, PK’, User certificate
No token exist
Upload data
CT, CK, PK’ AND data token
Token is saved
TOKEN
Verify ownership of user
User Signature, PK’
Verifying signature
Reencryted CK [if it is valid signature]
Dedupe record is added to corresponding data token
Reject request [Invalid signature]

Where CT = Encrypted user file

THE IIOAB JOURNAL ISSN: 0976-3104

CK = Key used for encryption is encrypted by dedupe checker's public key

PK'=Public key of user encrypted by dedupe checker's public key

Fig. 2: Deduplication procedure in Attribute and Policy based dedupe system

……

ARCHITECTURE DESCRIPTION

Initially, each CSP user should generate their personal credential such as user key pairs and get the certificate from the dedupe checker which is an authorized third party present in the cloud. To upload the data in CSP, the user should generate data token using hashing algorithm SHA 2. Then the user can request the dedupe checker for duplication check by sending data token, Public key of user encrypted by dedupe checker's public key [PK'] and user certificate.

The dedupe checker will verify the user certificate and then compare the data token with the tokens which is already present in its storage. If no tokens match with the token send by the user then the data is not present in the CSP. Now, the dedupe checker will request the user to upload the data. Then the user send encrypted data [CT], data token, PK' and CK [key used for encryption is encrypted by dedupe checker public key which is known as CK] to the CSP for storage. The CSP will forward only the token [since no subsequent upload of that data occurs] to the dedupe checker for storage.

If the token send by user matches with the existing token in dedupe checker then the duplication occurs. Then the dedupe checker will request the user to prove their ownership of data. The user will generate the signature based on their data token and user public key and send the signature and encrypted public key [using public key of dedupe checker] to dedupe checker for verification. The dedupe checker will validate the signature. If the signature is valid then the dedupe checker will reencrypt the CK and send it back to the user for accessing the data. Then a dedupe record will be add with the index of data token to the dedupe checker. If the signature is not a valid one then someone is trying to get the data from CSP by sending invalid signature. So, the dedupe checker will reject that user request.

LITERATURE SURVEY

This section gives a detailed review about various deduplication schemes. Here we reviewed how the deduplication problem is resolved in each scheme. While reviewing a scheme we listed the algorithms and techniques used in that scheme and the merit and demerit of that scheme are also specified. The following papers are survived in this section.

Deduplication on Encrypted Big Data in Cloud

It has a novel scheme for deduplication integrated with access control. It uses proxy re-encryption algorithm (1024 bit) for deduplication, AES (256 bit) for encrypting and decrypting a file, SHA 1 for hash function and ECC [18] (192 bit) for data ownership verification. This scheme also supports data sharing with deduplication even when the data holder is offline. We apply Elliptic curve cryptography and user certificate to prove the ownership of users. Initially to upload the data user will send a data token along with his pubic key and his user certificate. The cloud service provider (CSP) will verify the user certificate and check whether the data token already exist or not. If it doesn't exist then csp request the user to upload the data. If data token exist then csp will forward the user request to a trusted third party which is known as authenticated party (AP).The AP will challenge the user to prove that he/she contains the entire data. If the challenge is proved using ECC then a reencrypted key is provided to user to access the data. If else then the user request will be rejected by CSP. In this scheme Data Updating, Data Deletion and Data owner Management is also possible. This scheme has a merit that it resist offline brute force attack caused by convergent encryption. This scheme also saves storage space of CSP which leads to low cost for storage and it support access control on encrypted data. But the demerit in this scheme is same key is used for encrypting and decrypting a file which is not highly secure.

Encrypted Data Management with Deduplication in Cloud Computing

It proposes a method for deduplication along with secure access control using attribute based encryption [ABE]. It adopts AES (256 bit) for symmetric encryption, RSA for Public Key Cryptography (PKC) and CP-ABE [16] for data deduplication and SHA 1 for hash function. At first the user will create two pair of keys which are RSA key pair generation and ABE key pair generation. And the public keys of user must be certified by authorized third party. When user attempts to save data which is already present in CSP then the deduplication occurs. The user requests the CSP to upload the data by sending DATA PACKET. The data packet contains a cipher text of user (CT_u), cipher key of a user (CK_u), hash value of the text H (M), hash value signed by ABE secret key of a user H (M) by sk_u and the two certificates of user Cert (PK_u) & Cert (pk_u). CSP verifies the certificates then check whether the H (M) by sk_u is already present. If so then it is from same user it informs the user. If it is from different user then csp contact the data owner. The data owner will verify the eligibility of user. If the result is positive then data owner will issue a secret attribute for the user to access the data. And finally data owner report the CSP about the successful deduplication of a user then the CSP deletes the corresponding user cipher text and cipher key. In this method Data Updating, Data Deletion and Data owner Management is also possible. Some merits of this scheme are low operational and implementation cost and it does not depend on the third party for Key generation. But the disadvantage over here is it consume more time for key generation.

A Hybrid Cloud Approach for Secure Authorized Deduplication

It describes a scheme for authorized duplication check in hybrid cloud. In this scheme the private clouds maintain a table that contains user's public key and their privileges. It uses 256 bit AES algorithm for encryption, tag generation SHA-1 (256 bit) algorithm and for token generation HMAC [17] is produced using SHA-1. For data upload the user needs to contact private cloud to prove his/her identity. If the user proves then the private cloud find the privilege of the user. When user send the tag of the file to the private cloud it return the file token to the user. User will send the file token to the CSP for deduplication check. If the check is positive then the user has to that he/she is an authenticated data owner. It is done by running Proof of Ownership (PoW) on CSP. If it is proved then a pointer will be provided to user to access the file and Proof (i.e., signature on file token) & time stamp will also be provided. Now, the user uploads the privilege set for the corresponding file along with the proof from CSP to the private cloud. The private cloud first verifies the proof then it compute the file token and return the token to the user. If the check is negative then there is no need to run PoW. The same steps are followed and after the user get the file tokens from the private cloud he/she will compute the encrypted file by using convergent key and finally upload the file with its privilege. Only file uploading and retrieval is specified. This scheme is secured with insider and outsider attack which add merit to this scheme. But it has a disadvantage that files which are predictable are vulnerable to brute force attack.

A Scheme to Manage Encrypted Data Storage with Deduplication in Cloud

It uses Proxy re encryption (PRE) for data deduplication, AES for symmetric encryption and RSA for PKC. Here the user will create two pair of keys which are RSA key pair generation and PRE key pair generation. And the public keys of user must be certified by authorized third party (AP). When user attempts to save data which is already present in CSP then the deduplication occurs. The user requests the CSP to upload the data by sending DATA PACKET. The data packet contains a cipher text of user (CT_u), cipher key of a user (CK_u), hash value of the text H (M), hash value signed by PRE secret key of a user H (M) by sk_u and the two certificates of user Cert (PK_u) & Cert (pk_u). CSP verifies the certificates then check whether the H (M) by sk_u is already present. If so then it is from same user it informs the user. If it is from different user then CSP contact the AP. The AP will verify the eligibility of user. If the result is positive then AP will generate a reencryption key for the user to access the data and send the key to CSP. CSP forward the reencryption key to user and finally user report the CSP about the successful deduplication then the CSP records the deduplication information & deletes the corresponding user cipher text and cipher key. In this method Data Updating, Data Deletion and Data owner Management is also possible. This method is resilient to offline brute force and dictionary attack which becomes the advantage of this scheme. But this scheme cannot implement directly in cloud.

Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage

It has a scheme that uses randomized convergent encryption to manage dynamic changes in ownership of data. This scheme also uses AES with electronic code book algorithm for encryption and decryption, MD5 algorithm for key and token generation. Initially the cloud server will set up a binary tree for universal user to store their key encrypting key (KEK) which are used for the generation of group keys. Here users are at leaf node. The path from root to leaf is known as path keys. While encrypting a message user will find the hash value for message $K_i = H (M_i)$ and use that hash value for encrypting the key L which is used for encryption [$C2 = E_{ke} (L)$]. The hash value of key K_i is used as tag T_i. Encrypt the message that is to be uploaded in cloud by using key L ($C1 = E_L (M)$) and the cipher text is constructed using C1 and C2 [Ci = C1+ C2]. Now the user will upload $T_{i,}$ Ci $_{\&}$ ID of the user. Then user deletes the message and retains only its Key for access. The server will insert ID of the user in a group G_i and maintain the Ownership List L_i that store the tag value of that group. If Li is already exist then cloud will just insert the ID into G_i to avoid duplication. This is done when the user is a subsequent uploader. Then reencrypt the C1 by using a randomly selected group key (GK_i) which is C3. Then select the root node that covers the all the user of that group [KEK (G_i)] and encrypt the GK_i by root node keys E_k (GK_i) where k = KEK (G_i). Again the Ciphertext will be generated [C'i = C1'+ C2+ C3]. This ensures that only member of G_i can decrypt the cipher text. When user send tag and Id for access the data then cloud will check the tag is present in ownership list. If present then CSP will send Tag and C'$_{i.}$ Key update and data modification is also possible in this scheme. The merits are it is secure against the chosen-plaintext attack, collusion attack and poison attack. Is also ensures forward and backward secrecy of outsourced data. But it can't recover original data under data loss attack because all duplicates are removed from the CSP.

Encrypted Data Deduplication in Cloud Storage

It proposes a scheme in which the user construct a cipher structure that contains four blocks namely check block, converting block, enabling block and cipher block. In this scheme SHA 2 algorithm is used for hash function, AES 128 bit algorithm is used for Symmetric encryption, RSA algorithm is used for Asymmetric encryption and Elgamal algorithm is used for homomorphic encryption. First, the hash value of the file is calculated and it is saved as Check Block where as Cipher Block contains the encrypted file which is encrypted using AES key shorter than prime modulo p. Then the encrypted AES key by user public key is saved as Enabling Block and it is encrypted using multiplicative asymmetric homomorphic encryption. At last the user computes the second hash value of the file and it is multiplied by the AES key under the modulus p which is known as Converting Block. Then the cipher structure will be uploaded to the cloud by user. The cloud server will identify the duplicate files by verifying the Cipher block of the structure. If there is no match found in the CSP then the whole cipher structure is saved in the CSP. If the match exists then CSP will convert the Enabling Block by encrypting the AES key of user1 by public key of user2 and CSP saves only the Converted Enabling Block rather than saving the cipher structure uploaded by user2. In this situation when user2 request for data retrieval then CSP will send the converted Enabling Block along with the cipher block of user1.Then the user decrypt the converted Enabling Block by using his/her secret key to obtain AES key. Finally by using AES key the user2 can decrypt the cipher block. This scheme also specifies only the file uploading and retrieval. The merit of this scheme is it improves the security of the cipher text by using randomized encryption algorithm. But it has a demerit too that is the total cost of this scheme is larger than the existing system [19] [20] [21] [22]

HEDup: Secure Deduplication with Homomorphic Encryption

It describe about the deduplication will take place with the help of Key Server deployed at CSP. Key Server is mainly used to provide Data Encryption Key to the users. Here AES 128 bit algorithm is used for symmetric encryption and MD5/SHA algorithms can be used for hash functions. When different users upload same file then same data Encryption Key will be provided by Key Server. In this scheme I the file uploading scenario is classified into two types namely First upload and subsequent upload. In First upload the user will authenticate with the Key Server using their credentials. If it is Success then key server will generate public and private key for the user using asymmetric encryption. Then the user will encrypt the file using his/her public key and send it to the key server for duplication check. If the Key Server returns Null then the client will send hash of the file H(F) and hash part of the file encrypted by R [random number] [E(H(F'), R]. Now the Key Server will save the H (F) as Key and [E (H (F'), R] as value. At last the client authenticate the storage [CSP]and save the cipher text C1 and cipher Key C2.If it is a subsequent upload then during duplicate check the hash value will be present in the Key Server. And Key server will return S = E (H (F'), R) to the user. Only valid user can obtain R from the S. After obtaining R the user will authenticate with storage and save their C1 and C2. The only difference in Scheme II is during subsequent upload the clients no need to calculate C1. The Key server will provide link of C1 to storage after the client authenticate with storage. In scheme III the homomorphic XOR [14] operation is included to enhance the security. In this three schemes only file uploading and retrieval is specified. The advantage is it resists man in middle attack since we are encrypting the file in client side. The disadvantage is vulnerable to passive attack such as unauthorized reading of the file and traffic analysis etc.

BDO-SD: An Efficient Scheme for Big Data Outsourcing with Secure Deduplication

It says about outsourcing big data with data deduplication. It is done using convergent encryption and it also has Keyword search over encrypted data. This method support both file level and block level deduplication. It adopt AES (256 bit) for symmetric encryption, SHA 1 for hash function, blind signature system [13] is used for data owner verification. Initially each user will register with trusted authority (TA) and it sets (gxj, xj) be user public key and private key. And for each data owner TA will assign ID-based Key pairs. And data owner generate the convergent keys for file by using blind signature system. Then the data owner generates tag [12] for the file and sends it to CSS. CSS verify that tag is already present. If yes then data owner should pass the ownership authentication. If it is a valid data owner then a pointer for the file will be sent to the owner and CSS will abort the upload process. If result of deduplication check is no then block level deduplication will be preceded. The user will generate tag for blocks and send to CSS. If tag matches then ownership of user is verified. If it is valid user then pointer of blocks will be saved. If tag for blocks doesn’t match then cipher block is generated using convergent keys. For that purpose, encrypted convergent keys should be generated for each block. Using encrypted convergent keys keyword search is possible. Then data owner uploads the unique blocks {Bi}, all encrypted blocks and encrypted keys (Ci, CKi, Si, CK’i), as well as T (Bi) to the CSS. Then CSS will store the signature, cipher key and tag for blocks. If CSS receive the request for retrieval then it verify that user is eligible for retrieval after verification the cipher text and encrypted convergent keys will be provided to user. In this scheme trapdoor generation is also specified. The merit is it resists brute force attack. And the demerit is this scheme has more communication overhead.

Secure Deduplication of Encrypted Data without Additional Independent Servers

It briefly describe about cross-user deduplication scheme that supports client-side encryption without requiring any additional independent servers. It uses password authenticated key exchange [PAKE] protocol. PAKE Protocol is used for deduplication, AES algorithm is used for pseudo random function and elgamal algorithm is used for additive homomorphic encryption. To upload a file the client C will calculate short hash and hash value for the file. The short hash may be same for many different files. When client send short hash to storage it check for existing clients who have uploaded files with same short hash. Then client should run the PAKE protocol with hash as input function. Then client get the session keys of existing user who upload the same file. Then the existing clients & C use the pseudorandom function to extend the length of session key and split the key into two one is left part of key and other is right part of key. Then left part of key, client public key, and encrypted right part of the key is sent to storage from the existing client and C. then storage check the index value and use homomorphic encryption to calculate e send e to client. Then client calculate the key K_f from e and encrypt it then it send to the storage. If it is already present in the storage S then s will pointer pointer for the client to access the data. Here the pseudo random function is implemented using AES 128 bit algorithm and homomorphic encryption is done using Elgamal algorithm. Online brute force attack can be prevented which add a merit to this scheme. It is vulnerable to offline brute force attack, dictionary attack and hacking of low level entropy passwords and if block level deduplication takes place then it will cause additional overhead.

Enhanced Secure Thresholded Data Deduplication Scheme for Cloud Storage

It deals with the new concept known as data popularity. It describes an encrypted scheme that support secure cipher text of a file is downgraded to convergent cipher text of file that support deduplication. It happens when the file become popular. Here two trusted authority is used one is Identity Provider and the other is index repository service [IRS]. Here AES – 128 – CTR used as symmetric encryption, SHA 256 is used for hashing function. The user encrypts the file to generate index and sent it to IRS. If IRS returns the index unchanged then the file is already popular. Then only user is added to the owner list. If IRS return different index then the data is unpopular. Then doubly encrypted file with encrypted random key is send to storage. In this scheme deduplication request is sent to storage using IRS. IRS will send indexes and decryption shares to storage. Then x check the corresponding record for the indexes and decrypt the records using decryption shares. If all the convergent cipher text is equal then it add a new record under index and delete the copies of convergent cipher text. If it is not equal then some clients have cheated S so it abort the deduplication process and sent a failed message to IRS. This scheme also support file retrieval and deletion from the storage. The merit of this scheme is it has low cost and resists user collusion attack. The demerit is it cannot prevent attacks based on knowledge of the hash of plaintext file.

PERFORMANCE ANALYSIS

This section illustrates the comparison between efficiency of encryption and decryption process present in the existing scheme [2] and the proposed work.

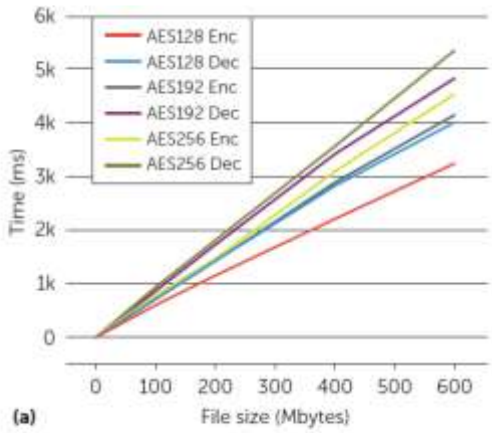
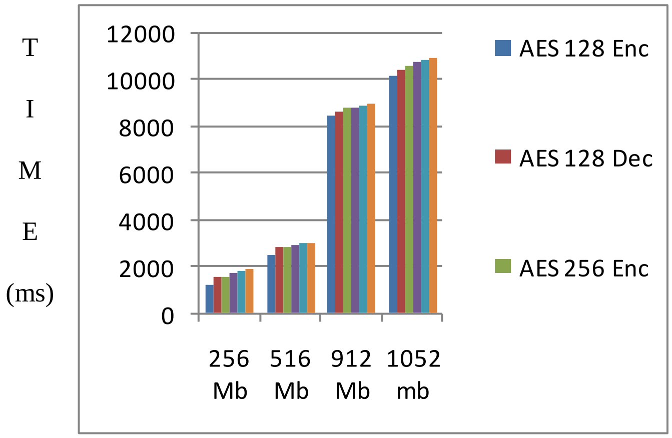


Fig. 3: Operation time of encryption and decryption using AES

……



b) File size (Mbytes)

Fig. 4: Operation time of Encryption and Decryption of AES and Attribute and Policy based system

……

Since all the existing schemes assume that the symmetric encryption is secure and they encrypted the user file using AES algorithm. So, we have analyzed the efficiency of different types of AES in **Figure- 3**. Among the three types of AES, AES 256 is secure and efficient.

In our proposed system the encryption and decryption is based on attribute and policy. So we make a comparison between AES and Attribute and Policy based system in **Figure- 4**. Compared to AES 256, Attribute and policy based system is highly secure. The time taken for encryption and decryption increases with increase in data size. But it takes only few milliseconds. So, Attribute and policy based system is efficient and provides high security than the existing schemes.

CONCLUSION

In this paper, we have survived various deduplication schemes which eliminate the replicates of files from the cloud and also we analyzed those schemes to find the finest method for deduplication. Based on the analysis, many demerits were found which reduce the efficiency and security of deduplication process. To overcome these limitations we have proposed Attribute and Policy based dedupe system to enhance the security of deduplication schemes and also it provide security against tampering, unauthorized access etc. In future, the proposed method will be implemented with additional features such as data updation, data deletion in cloud etc., which increase the security & efficiency of deduplication process higher than the existing schemes.

CONFLICT OF INTEREST

The authors declare no conflict of interests.

ACKNOWLEDGEMENT

None

FINANCIAL DISCLOSURE

None

REFERENCES

[1] Zheng Yan, Wenxiu Ding, Xixun Yu, Haiqi Zhu, and Robert H. Deng. [2016] Deduplication on Encrypted Big Data in Cloud," in IEEE TRANSACTIONS ON BIG DATA, 2(2) APRIL-JUNE pp. 138- 150.

[2] Zheng Yan, Mingjun Wang, Yuxiang Li, and Athanasios V. Vasilakos, Encrypted Data Management with Deduplication in Cloud Computing," in IEEE Cloud Computing, March/April 2016, pp. 29- 35.

[3] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, and Wenjing Lou, "A Hybrid Cloud Approach for Secure Authorized Deduplication," in IEEE Transactions on Parallel and Distributed Systems.

[4] Zheng yan, Wenxiu Ding, and Haiqi Zhu, "A Scheme to Manage Encrypted Data Storage with Deduplication in Cloud," in Springer International Publishing Switzerland 2015, pp. 547–561.

[5] Junbeom Hur, Dongyoung Koo, Youngjoo Shin, and Kyungtae Kang, "Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage," in IEEE Transactions on Knowledge and Data Engineering, 1041-4347 (c) 2016 IEEE.

[6] Chun-I Fan, Shi-Yuan Huang and Wen-Che Hsu in "Encrypted Data Deduplication in Cloud Storage," in 2015 10th Asia Joint Conference on Information Security, pp. 18-25, 978-1-4799-1989-5/15 $31.00 © 2015 IEEE.

[7] Rodel Miguel, Khin Mi Mi Aung, and Mediana, "HEDup: Secure Deduplication with Homomorphic Encryption," 978-1-4673-7891-8/15/$31.00 ©2015 IEEE, pp. 215 – 223.

[8] Mi Wen, Kejie Lu, Jingsheng Lei, Fengyong Li, and Jing Li, "BDO-SD: An Efficient Scheme for Big Data Outsourcing with Secure Deduplication," in The Third International Workshop on Security and Privacy in Big Data (Big Security2015), pp. 214 – 219.

[9] Jian Liu, N. Asokan, and Benny Pinkas, "Secure Deduplication of Encrypted Data without Additional Independent Servers," pp. 874 – 885.

[10] Jan Stanek, and Lukas Kencl. [2012] Enhanced Secure Thresholded Data Deduplication Scheme for Cloud Storage, in IEEE Transactions On Dependable And Secure Computing, 1545-5971 (c) IEEE.

[11] http://www.druva.com/blog/understanding-data-deduplication/

[12] JR Douceur, A Adya, WJ Bolosky, D Simon, M Theimer.[2002] Reclaiming Space from Duplicate Files in a Serverless Distributed File System", in Proceeding of ICDCS, pp. 617-624.

[13] M Bellare, S Keelveedhi. [2013]DupLESS: Server-aided encryption for deduplicated storage," in Proceeding of USENIX Security Symposium, pp. 179-194.

[14] Shu Qin, Ren., et al. "Homomorphic Exclusive-Or Operation Enhance Secure Searching on Cloud Storage," in Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on 2014.

[15] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in Proc. Cryptology—EUROCRYPT, 2013, pp. 296–312, doi:10.1007/978-3-642-38348-9_18.

[16] http://crypto.stackexchange.com/questions/17893/what-is attri bute-based-encryption.

[17] http://searchsecurity.techtarget.com/definition/Hash-based Mes sage -Authentication-Code-HMAC

[18] https://bithin.wordpress.com/2012/02/22/simple-explanation-for-elliptic-curve-cryptography-ecc/

[19] J Daemen, V Rijmen.The advanced encryption standard (AES)," United States National Institute of Standards and Technology (NIST), vol. Federal Information Processing Standards Publication 197, November 26 2001,http://csrc.nist.gov/publications/fips/ fips197/ fips-197.pdf.

[20] NSA. (NSA), "Secure hash standard (SHS)," United States National Institute of Standards and Technology (NIST), vol. Federal Information Processing Standards Publication 180-4, March 2012,http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf.

[21] R Rivest, A Shamir, L Adleman.[1978] A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, 21(2):. 120126,, http://people.csail.mit.edu/rivest/Rsapaper.pdf.

[22] T Elgamal.[1985] A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Trans. on Information Theory, 31(4): 469–472, 1985.