Deduplication of Encrypted Bigdata on Cloud

Srikanth M S[1], Anusha Kamath[2], Battula Sheetal[3], Meghana M Kalyan[4], Meghana S[5]

[1]Assistant Professor, Dept. of Computer Science Engineering, Sapthagiri College of Engineering, Karnataka, India
[2,3,4,5]Student, Dept. of Computer Science Engineering, Sapthagiri College of Engineering, Karnataka, India

---***---

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. In storage services with huge data, the storage servers may want to reduce the volume of stored data, and the clients may want to monitor the integrity of their data with a low cost, since the cost of the functions related to data storage increases proportionally with the size of the data. In order to preserve the privacy of data holders, data are often stored in cloud in an encrypted form. To achieve these goals, secure deduplication and integrity auditing delegation techniques are required, which can reduce the volume of data stored in storage by eliminating duplicated copies and permit clients to efficiently verify the integrity of stored files by delegating costly operations to a trusted party. However, encrypted data introduces new challenges for cloud data de duplication, which becomes crucial for big data storage and processing in cloud. Existing solutions of encrypted data de duplication suffer from security weakness. In this paper, we propose a scheme to de duplicate encrypted data stored in cloud based on ownership challenge and BLS signature based homomorphic linear authenticator. We evaluate its performance based on extensive analysis and computer simulations. The results show the superior efficiency, satisfies all the fundamental security requirements. and effectiveness of the scheme for potential practical deployment, especially for big data de duplication in cloud storage.*

***Key Words*: de-duplication, proxy re-encryption, cloud storage, big data de duplication**

1. INTRODUCTION

Cloud computing offers a new way of Information Technology services by rearranging various resources (e.g., storage, computing) and providing them to users based on their demands. Cloud computing provides a big resource pool by linking network resources together. It has desirable properties, such as scalability, elasticity, fault-tolerance, and pay-per-use. Thus, it has become a promising service platform. The most important and popular cloud service is data storage service.

Larger volumes of data require higher cost for managing the various aspects of data, since the size of data influences the cost for cloud storage services. The scale of storage should be increased according to the quantity of data to be stored. In this viewpoint, it is desirable for storage servers to reduce the volume of data, since they can increase their profit by reducing the cost for maintaining storage. On the other hand, clients are mainly interested in the integrity of their data stored in the storage maintained by service providers. To verify the integrity of stored files, clients need to perform costly operations, whose complexity increases in proportion to the size of data. To reduce the volume of data, deduplication has to be performed in servers so that the storage space efficiency can be improved by removing duplicated copies.

Cloud users upload data to the data center of a Cloud Service Provider (CSP) and allow it to maintain these data. Since intrusions and attacks towards sensitive data at CSP are not avoidable, it is prudent to assume that CSP cannot be fully trusted by cloud users. Moreover, the loss of control over their own personal data, leads to high data security risks, especially data privacy leakages. Due to the rapid development of data mining and other analysis technologies, the privacy issue becomes serious. Hence, a good practice is to only outsource encrypted data to the cloud in order to ensure data security and user privacy.

But the same or different users may upload duplicated data in encrypted form to CSP, especially for scenarios where data are shared among many users. We have designed a new scheme for secure and efficient cloud storage service. The scheme supports both secure deduplication and integrity auditing based on the homomorphic linear authenticator (HLA), which is designed using BLS signature in a cloud environment. The proposed scheme provides secure deduplication of encrypted data and is more efficient to support deduplication and public auditing at the same time. Consequently, de duplication becomes critical for big data storage and processing in the cloud.

1.1 Overview of Present Work

Cloud storage services have become widespread, and there is an increase in the use of cloud storage services. cloud storage services will increase due to the performance of the new networking technique. In this viewpoint, we can characterize the volume of data as a main feature of cloud storage services. Many service providers have already prepared high resolution contents for their service to utilize faster networks. For secure cloud services in the new era, it is important to prepare suitable security tools to support this change. Larger volumes of data require higher cost for managing the various aspects of data, since the size of data influences the cost for cloud storage services. The scale of storage should be increased according to the quantity of data to be stored.

1.2 Problem Statement

Clients are mainly interested in the integrated of their data stored in the storage maintained by service providers. To verify the integrity of stored files, clients need to perform costly operations, whose complexity increases in proportion to the size of data.

2. LITERATURE SURVEY

[1] Efficient Client-Side Deduplication of Encrypted data with Public Auditing in Cloud Storage

As there is a considerable increase in the amount of data stored in storage services along with rapid evolution of the networking techniques. Here Secure deduplication and integrity auditing delegation techniques have been studied to reduce the volume of data stored in storage by eliminating duplicated copies and permits client to efficiently verify the integrity of stored files. The combined is implemented to perform both secure deduplication of encrypted data and public integrity auditing of data. To reduce the volume of data, deduplication has to performed in servers so that the storage space efficiency can be improved by removing duplicated copies and almost the 75% of the data are duplicated, this fact raises the need for design of deduplication technology. To support the two functions, this scheme performs response protocols using the BLS Signature based on homomorphic linear authenticator. This scheme consists of the entities like client, cloud storage server and TPA (Third Party Auditor) to support public integrity auditing. Following figure shows the system model containing the entities.

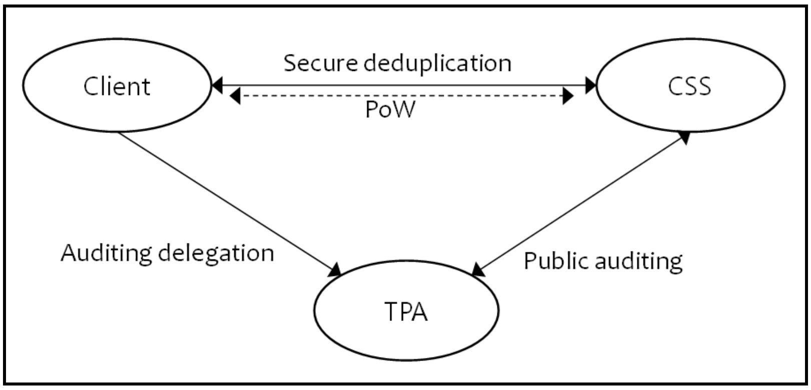

Figure 1: System model

The proposed scheme satisfies the following objectives:

1. Privacy: Except for the information about duplication, no information about the outsourced data is revealed to an adversarial party.
2. Secure deduplication: Secure deduplication is supported without revealing any information except for the information about duplication.
3. Public verifiability*:* The TPA is able to examine the accuracy and availability of the outsourced data without querying the entire data and without intervention by the data owner.
4. Storage correctness*:* If the CSS is keeping the user's data intact, it can pass the TPA's verification.

As the scheme satisfies the security objectives by providing the deduplication for the ciphertext and performing PoW protocol and from the privacy perspective, the proposed method outsources the encrypted ciphertext to the CSS using the convergent encryption key. In the integrity auditing process, the TPA can also partially obtain the information about the ciphertext.

When storing data on remote cloud storages, users want to be assured that their stored data are maintained accurately in the storage without being corrupted. In addition, cloud servers want to use their storage more efficiently. To satisfy both the requirements, the scheme prevents the leakage of important information about user data and simultaneously supporting public auditing of encrypted data.

[2] Secure Data Deduplication in Cloud

Deduplication is a process that eliminates redundant copies of data and reduces storage required to store the duplicated data. Data deduplication becomes more and more a necessity for cloud storage providers because of the continuous and exponential increase in the number of users and the size of their data. The privacy of data holders is more important, so in order to preserve the privacy of the data holders, data is stored in the cloud in an encrypted form. Here the scheme is implemented to deduplicate the encrypted data which is stored in the cloud and also, we use RAS and AES algorithm for encryption process. The simple idea behind deduplication is to store duplicate Data only once. Unfortunately, deduplication and encryption are two conflicting technologies. While the aim of deduplication is to detect duplicate data and store them only once, the result of encryption is to make two identical data indistinguishable after being encrypted. Convergent encryption is a technique which has been proposed to meet these two conflicting requirements. Here the scheme is proposed to deduplicate encrypted data at CSP by applying PRE to issue keys to different authorized data holders based on data ownership challenge. It is applicable in scenarios where data holders are not available for deduplication control. The scheme also addresses every problem and provides a secure convergent encryption for efficient encryption without considering issues of the key management and block level deduplication. And the scheme involves the entities such as Data Owner, Cloud server, TPA and End User. And below figure shows the participation of each and every entity in uploading and accessing a file which is stored in storage.

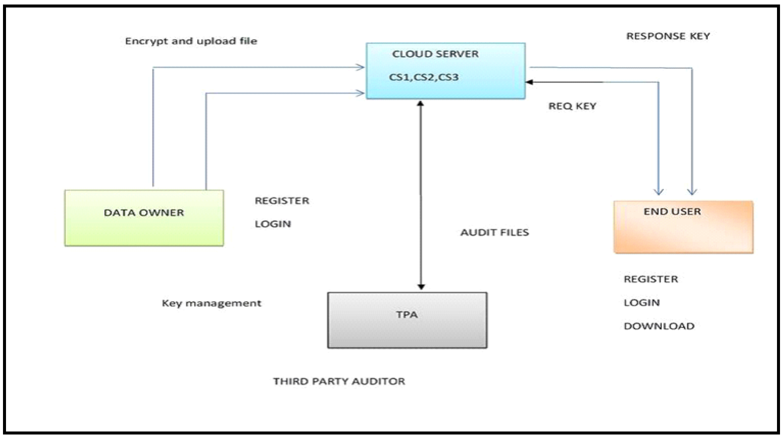

Figure 2. System model

Advantages

- The client is permitted to perform the duplicate copy check for records selected with the particular subject.
- The complex subject to help stronger security by encoding the record with distinct privilege keys.
- Decrease the storage space of the tags for a reliability check. To strengthen the security of deduplication and ensure the data privacy.
- Any adversary could not directly derive the convergent key from the content of the file and the dictionary attack is prevented.
- Any adversary without the file can-not convince the cloud storage service to get the corresponding access privilege

Managing encrypted data with deduplication is most significant in practice for running a cloud storage service which is secure and dependable, especially for data processes.

[3] DeDu: Building a Deduplication Storage System over Cloud Computing

This paper provides the paradigms of deduplication storage system over cloud computing. The simulation results demonstrate that our deduplication cloud storage system is more efficient than traditional deduplication approaches. Theories and Approaches used in the above paper by The SUN and Jianming YONG is as follows:

a. Identifying the duplication

The duplication of data in cloud can be done in two ways. One is comparing blocks or files bit by bit and the other is comparing the blocks by hash values. In comparing the blocks, the advantage is that it is accurate but it consumes lot of time. In hash the advantage is that it is fast but there can be collision.

b. Storage Mechanism

Two storage mechanisms can be used to access the data access requirements. One is used to store mass data and the other one is used to store index. Several data baes such as Oracle, SQL, LDAP, BigTable are used as index symbols.

System Design of the proposed paper:

a. Data Organization

Here the HDFS and HBASE must collaborate to guarantee that the system is working well. In this approach we separate the files and link the files into different folders. In DeDu system, each source file is read by its primary value and saved in a folder. HBase records all the hash values for each file, there is only one table in HBase, which is names as "DeDu"

b. Storage of the files

This has three main steps to save a file. First is to hash a value to a client; Second od to identify any duplication; third is to save the file.

HDFS will store source files, which are uploaded by users, and corresponding link files, which are automatically made by DeDu and record the source file's hash value and the logical path of the source file.

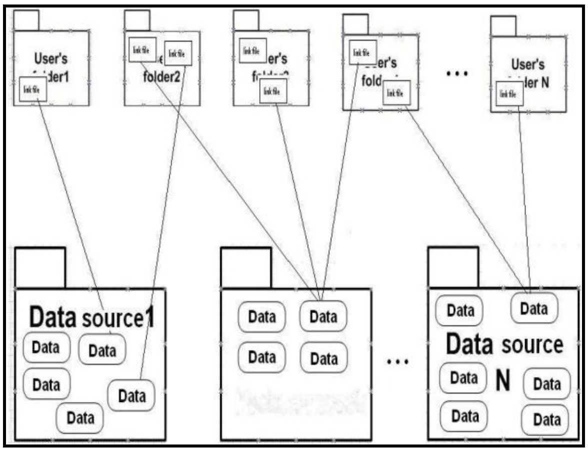

Figure 3: Data organization.

c. Access to the files

We can use link file as a way to access the files. Each link records two things; the hash value and the logical path to the source file.

d. Deletion of Files

This referred paper has proposed two models for deletion; in one case, the file is pseudo-deleted and in other case, it is fully-deleted. The advantage is that by using scalable and parallel deduplicated storage named DeDu for cloud environment, it is not only useful for the organizations to store the data but also for the common users to store their private data. In this mentioned paper, hash value is calculated at the client side before transmitting the data. When duplicate data is found, real data transmission will not occur.

[4] Improved Proxy-Encryption Schemes with Applications to Secure Distributed Storage

This paper proposed by Distributed Storage, Giuseppe Ateniese, Kevin Fu, Matthew Green, Susan Hohenberger predicts fast and secure re-encryption for managing encrypted file systems. Proxy re-encryption allows a proxy to transform a cipher text computed under the sender's public key into the one that can be opened by the receiver using receiver's secret key.

Background of proxy re-encryption

A methodology for delegating decryption rights was first introduced by Mambo and Okamoto purely as an efficiency improvement over traditional decrypt-and-then-encrypt approaches.

Application of Proxy Re-encryption:

i. Secure File Systems:
Data is stored in files. To secure file system is a normal application of proxy re-encryption.

ii. Outsourced Filtering of Encrypted Spam:
Another application of proxy re-encryption is filtering the encrypted mails performed by authorized contractors.

1) Security of Unidirectional Proxy Re-encryption

A PRE scheme is represented as a tuple (KG; RG; E; R; D) (KG; E; D) are the standard key generation, encryption, and decryption algorithms. On input the security parameter 1^k, KG outputs a public and private key pair (pk_A ; sk_A) for entity A. On input pk_A and data M, E outputs a ciphertext CA = E (pk_A ; M). On input sk_A and ciphertext CA, D outputs the plain data M = D(skA;CA). On input (pk_A ; sk_A ; pkB), the re-encryption key generation algorithm RG, outputs re-encryption key rk_A->B for a proxy. On input rk_A->B and ciphertext CA, the re-encryption function R, outputs R(rk_A->B; CA) = E (pk_B ; m) = CB which can be decrypted with private key skB.

2) Symmetric Encryption

Encrypt (DEK; M). The Encrypt algorithm takes as input data M, the symmetric key DEK. It encrypts M with DEK and outputs the ciphertext CT. This process is conducted at user u to protect its data stored at CSP with DEK.

Decrypt (DEK; CT). The Decrypt algorithm takes as input the encrypted data CT, the symmetric key DEK. The algorithm decrypts CT with DEK and outputs the plain data M. A user (data holder) conducts this process to gain the plaintext of stored data at CSP.

[5] Encrypted Big Data

Cloud computing is used to mainly store data often in encrypted form. This encryption leads to new challenges as traditional deduplication techniques does not work efficiently on encrypted data. In this paper ownership challenge and proxy re-encryption techniques are used which tackles the above-mentioned problem in an efficient way.

Cloud computing provides a big resource pool by linking network resources together and has many desirable properties such as scalability, elasticity, fault-tolerance and pay-per-use. Cloud users upload personal and confidential data to the data center of a Cloud Service Provider (CSP). But the CSP cannot be fully trusted because there are chances of intrusions on sensitive data. Therefore, its better to encrypt data before uploading but we should also keep in mind to deduplicate this data so that we do not waste cloud storage.

Encrypted Data Upload*:* If data duplication check is negative, the data holder encrypts its data using a randomly selected symmetric key DEK in order to ensure the security and privacy of data, and stores the encrypted data at CSP together with the token used for data duplication check. The data holder encrypts DEK with pkAP and passes the encrypted key to CSP.

Data Deduplication: Data duplication occurs at the time when data holder tries to store the same data that has been stored already at CSP. This is checked by CSP through token comparison. If the comparison is positive, CSP contacts AP for deduplication by providing the token and the data holder's PRE-public key. The AP challenges data ownership, checks the eligibility of the data holder, and then issues a re-encryption key that can convert the encrypted DEK to a form that can only be decrypted by the eligible data holder.

Data Deletion*:* When the data holder deletes data from CSP, CSP firstly manages the records of duplicated data holders by removing the duplication record of this user. If the rest records are not empty, the CSP will not delete the stored encrypted data, but block data access from the holder that requests data deletion. If the rest records are empty, the encrypted data should be removed at CSP. Data Owner Managemen*t:* In case that a real data owner uploads the data later than the data holder, the CSP can manage to save the data encrypted by the real data owner at the cloud with the owner generated DEK and later on, AP supports re-encryption of DEK at CSP for eligible data holders.

Encrypted Data Update: In case that DEK is updated by a data owner with DEK0 and the new encrypted raw data is provided to CSP to replace old storage for the reason of achieving better security, CSP issues the new re-encrypted DEK0 to all data holders with the support of AP.

Scheme to Verify Data Ownership

Deduplication is checked using an ownership verification protocol based on a cryptoGPS identification scheme. An AP challenges the data holder to check if the user is an intruder or not. CSP checks for duplication by verifying with the token of the data that is already existing. If the same token exists, CSP passes the further verification to AP. If challenge verification is positive then AP generates re-encryption key issues it to CSP. CSP re-encrypts the key and provides it to the user.

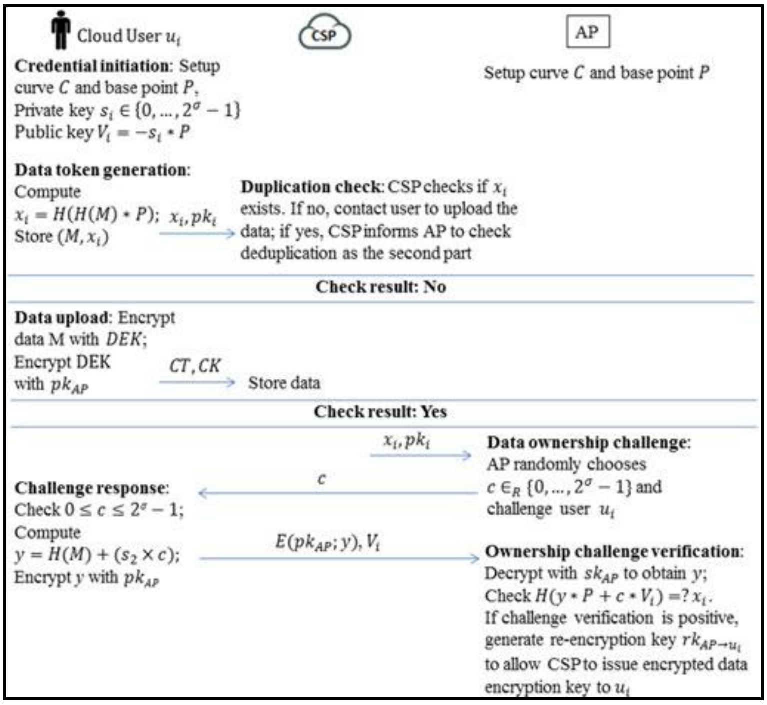
Cloud User u_i
CSP
AP
Credential initiation: Setup curve C and base point P,
Private key $s_i \in \{0,\dots,2^\sigma - 1\}$
Public key $V_i = -s_i * P$
Setup curve C and base point P
Data token generation:
Compute
$x_i = H(H(M) * P)$;
Store (M, x_i)
x_i, pk_i
Duplication check: CSP checks if x_i exists. If no, contact user to upload the data; if yes, CSP informs AP to check deduplication as the second part
Check result: No
Data upload: Encrypt data M with DEK;
Encrypt DEK with pk_{AP}
CT, CK
Store data
Check result: Yes
x_i, pk_i
Data ownership challenge:
AP randomly chooses $c \in_R \{0,\dots,2^\sigma - 1\}$ and challenge user u_i
c
Challenge response:
Check $0 \le c \le 2^\sigma - 1$;
Compute
$y = H(M) + (s_2 \times c)$;
Encrypt y with pk_{AP}
$E(pk_{AP}; y), V_i$
Ownership challenge verification:
Decrypt with sk_{AP} to obtain y;
Check $H(y * P + c * V_i) =? x_i$.
If challenge verification is positive, generate re-encryption key $rk_{AP \rightarrow u_i}$ to allow CSP to issue encrypted data encryption key to u_i

Figure 4. Data ownership verification.

Deduplication is checked using an ownership verification protocol based on a cryptoGPS identification scheme. An AP challenges the data holder to check if the user is an intruder or not. CSP checks for duplication by verifying with the token of the data that is already existing. If the same token exists, CSP passes the further verification to AP. If challenge verification is positive then AP generates re-encryption key issues it to CSP. CSP re-encrypts the key and provides it to the user.

Procedures

The deduplication of data happens as follows:
Step 1: The data token is generated by using the algorithm
Step 2: The CSP checks for duplicate data by comparing the generated token with the tokens that are already present.
Step 3: If the token matches then data isn't saved but a pointer is made to point to the same data. If not, then the data is uploaded.
Step 4: When the user requests to access this data the ownership challenge is posed. If the user gives proper verification then he is allowed to access the data and download it.

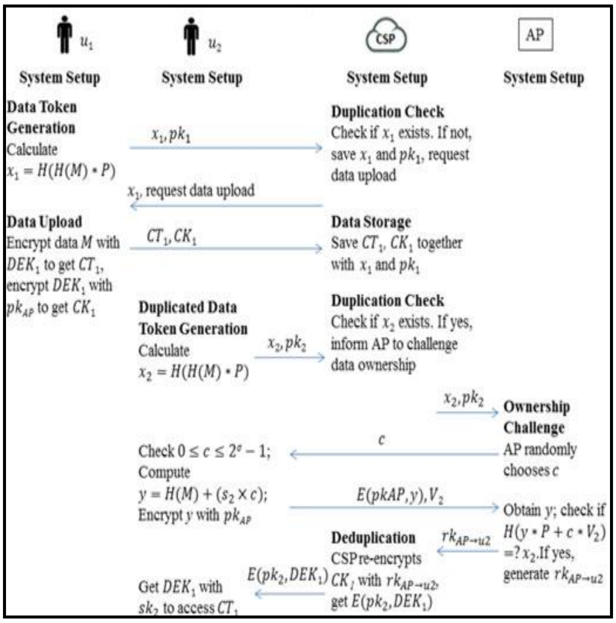
u_1
u_2
CSP
AP
System Setup
System Setup
System Setup
System Setup
Data Token Generation
Calculate
$x_1 = H(H(M) \cdot P)$
x_1, pk_1
Duplication Check
Check if x_1 exists. If not, save x_1 and pk_1, request data upload
x_1, request data upload
Data Upload
Encrypt data M with DEK_1 to get CT_1, encrypt DEK_1 with pk_{AP} to get CK_1
CT_1, CK_1
Data Storage
Save CT_1, CK_1 together with x_1 and pk_1
Duplicated Data Token Generation
Calculate
$x_2 = H(H(M) \cdot P)$
x_2, pk_2
Duplication Check
Check if x_2 exists. If yes, inform AP to challenge data ownership
x_2, pk_2
Ownership Challenge
AP randomly chooses c
c
Check $0 \leq c \leq 2^{\sigma} - 1$;
Compute
$y = H(M) + (s_2 \times c)$;
Encrypt y with pk_{AP}
$E(pkAP, y), V_2$
Obtain y; check if $H(y \cdot P + c \cdot V_2) =? x_2$.If yes, generate $rk_{AP \to u2}$
$rk_{AP \to u2}$
Deduplication
CSP re-encrypts CK_l with $rk_{AP \to u2}$, get $E(pk_2, DEK_1)$
$E(pk_2, DEK_1)$
Get DEK_1 with sk_2 to access CT_1

Figure 5. A procedure of data deduplication

- Data Deletion at CSP
  If a user requests to delete data then his access to this data is removed by the CSP. Later the deduplication record is checked. If it is empty then related records are deleted as well.
- Data Owner Management
  In case the data holder uploads the data before the data owner then the owner should complete certain challenges by providing certificates to prove his claim.
- Encrypted Data Update
  If the user requests to update data then the CSP checks for validity and allows the user to update.

Managing the data in an efficient and secure manner is very important. In this paper efficient management of storage on cloud and in a secure manner through encryption has been proposed.

[6] Flexible Data Access Control Based on Trust and Representation in Cloud Computing

Cloud computing helps us in storing huge amount of data. For security purposes it is stored in the encrypted form. But the data may have to be accessed by other entities to fulfil an expected service. This can be done with the help of the data owner and certain trusted third-party service providers. The above described is done by applying Attribute-Based Encryption and Proxy Re-Encryption

Cloud computing provides a big resource pool by linking network resources together and has many desirable properties such as scalability, elasticity, fault-tolerance and pay-per-use. People nowadays perform various social activities such as texting, calling, etc. Trust is a very important aspect here. Social trust relationships can be established and assessed in a digital world. Trust and reputation play a decisive role in cyber security.

[7] ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage

The continuous increase in the number of users and the size of the data being stored, data deduplication becomes a necessity for cloud storage providers. By storing a single copy of duplicate data, cloud providers reduce most of their storage and the data transfer costs. In this we are using a secure and efficient storage service where deduplication takes place at the block level and there is confidentiality of data at the same time. Convergent Encryption makes sure that deduplication and confidentiality takes place. But the drawback of this is that it doesn't ensure protection of predictable files against dictionary attacks. To overcome this issue, they have proposed to add a secret value S to the encryption key.

Deduplication will be applied only to the files of those users who share the secret. The encryption key is defined as K = H(S|M) , where | denotes an operation between S and M.

However, this solution overcomes the weaknesses of convergent encryption. This approach provides data confidentiality without impacting deduplication effectiveness. Therefore, we propose ClouDedup, which does not depend on the security of one single component and it manages block-level deduplication in an effective manner.

The components of the referred paper are,

1. The server: The core component of ClouDedup is a server that implements the additional encryption operation to cope with the weaknesses of Convergent Encryption. 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.
2. Block-level Deduplication and Key Management*:* The requirement for deduplication at block-level raises an issue with respect to key management. In case of block-level deduplication, the requirement to memorize and retrieve CE keys for each block in a secure way, needs fully-fledged key management solution. Therefore, we suggest to include a new component, the metadata manager (MM), in the new ClouDedup system in order to implement the key management.
3. 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. Server is a trusted component with respect to user authentication, access control and additional encryption. 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 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. Among the potential threats, we also identify 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. 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.
4. Security: When server has applied the additional encryption, data is no longer vulnerable to CE weaknesses. 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 encryption. The primary role of the server is to retain the secret key used for the additional encryption. In real time, this goal can be effectively accomplished by using a hardware security module. When data is 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.

3. PROPOSED WORK

We proposed a scheme to achieve both secure deduplication and integrity auditing in a cloud environment. To prevent leakage of important information about user data, the proposed scheme supports a client-side deduplication of encrypted data, while simultaneously supporting public auditing of encrypted data. We used BLS signature based homomorphic linear authenticator to compute authentication tags for the PoW and integrity auditing. The proposed scheme satisfied the security objectives, and improved the problems of the existing schemes. In addition, it provides better efficiency than the existing schemes in the viewpoint of client-side computational overhead. Finally, we designed two variations for higher security and better performance.

ADVANTAGES

- Higher security.
- Better efficiency and performance.
- To prevent leakage of important information about user data.

4. METHODOLOGY

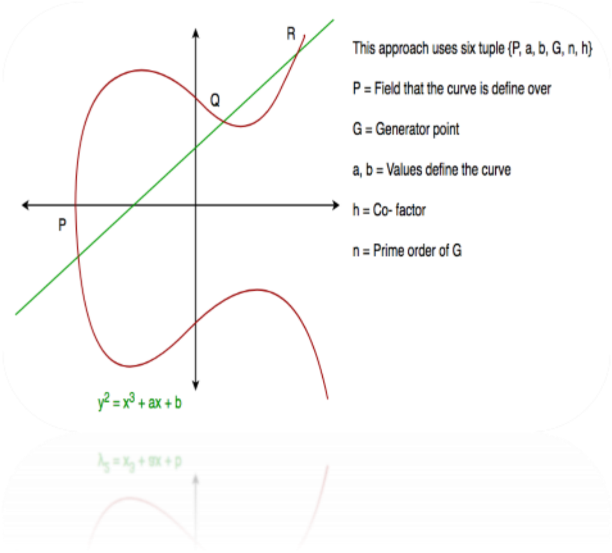
R
Q
P
This approach uses six tuple {P, a, b, G, n, h}
P = Field that the curve is define over
G = Generator point
a, b = Values define the curve
h = Co- factor
n = Prime order of G
$y^2 = x^3 + ax + b$

Fig 7: ECC curve

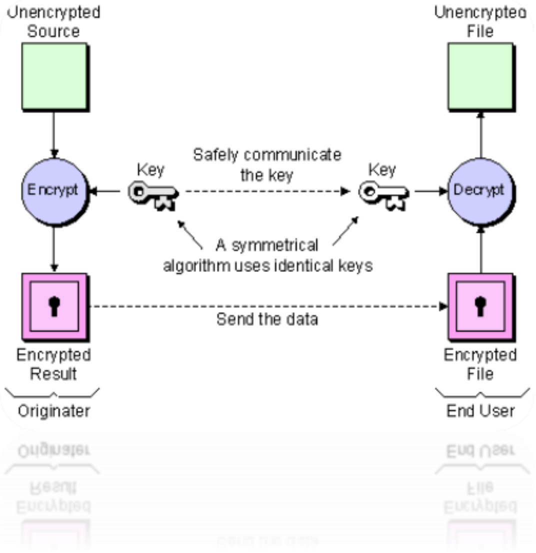

Fig 8: Encryption and Decryption

ECC (elliptic curve cryptography) technique will be used for encryption and decryption of data.

The following are symbols we are used,

E-->Elliptic curve

P-->Point on the curve

n-->/Maximum limit (prime number)

Generation of keys:

Keys will be used for the encryption and decryption. Here we are using public key for encryption and private key for decryption. We have to choose a number 's' within range of 'n'. Using following formula, we can generate the public key

→W=s*p

Where s = the random number selected within the range (1 to n-1)

P is point on curve.

→'W' is public key and 's' is private key.

Encryption

Assume 'x' is the data that sensed by the sensor sent to the Geo-social network. Represent this data on curve. Consider 'x' as point 'M' on the curve 'E'. Randomly select 'k' from [1-(n-1)]. Two cipher texts will be generated let be m1 and m2

M1 = k * p

M2 = M + k * W

Decryption:

We have to decrypt the data send by senor

X = M2 – s * M1;

Where x is the original message.

Proof:

X = M2 – s * M1

'x' can represent as 'M2 – s * M1'

M2 – s * M1 = (x + k * Q – s * (K * P))

(M2 = x + K * Q & M1 = K * p) = X + k * s * P – S * K * p (cancel k*s*p)

=x (original message)

Over All Result:

Deduplication is an avoiding repeated file stored to the cloud. Cloud having boundaries in size. So, without checking to store a file to cloud is burdening. This Project gives the idea and managing memory space in to the cloud. Obviously if no confusion in cloud to search the file. So, processing speeds competitively prior.

For the Security Are using file mechanism and the security purpose there are many algorithms are used. But in ECC it will help full for small key is used and time generation for key is less and performance is quite good in this algorithm.

ECC AND RSA

RSA

RSA is considered as the first real life and practical asymmetric-key cryptosystem. It becomes de facto standard for public-key cryptography. Its security lies with integer factorization problem. RSA's decryption process is not efficient as its encryption process. Many researchers have proposed to improve the efficiency of RSA's decryption using Chinese Remainder Theorem (CRT). Verma et al. proposed a model to improve decryption time of the RSA using CRT. They also proposed to generate large modulus and cryptographic keys with small order of a matrix.

For better and stronger security of data, bigger key sizes require, which means more overhead on the computing systems. Nowadays small devices are playing an important role in the digital world, which has less memory but needs security to cope with market demand. In this scenario, RSA becomes second thoughts.

RSA Algorithm
Key Generation

Step I. Select p, q p and q both are primes, $p \neq q$

Step II. Calculate $n = pq$

Step III. Calculate $\Phi(n) = (p - 1)(q - 1)$

Step IV. Select integer e $\gcd(\Phi(n), e) = 1$; $1 < e < \Phi(n)$

Step V. Calculate d $d \equiv e$-1 (mod $\Phi(n)$)

Step VI. Public key PU = $\{e, n\}$

Step VII. Private key PR = $\{d, n\}$

Encryption
Step I. Plaintext: $M<n$

Step II. Ciphertext: $C=Me$ mod n

Decryption

Step I. Ciphertext: *C*

Step II. Plaintext: *M*=*Cd* mod *n*
Here, key generation is to be done by each party, so that they can communicate each other securely. In the RSA algorithm, 'e' is for encryption, should be chosen such that gcd(Φ(n), e) is equal to 1. Once 'e' is selected, corresponding, 'd' that is for decryption should be generated with the help of finding the inverse of 'e' mod Φ(n).
In encryption process, a sender has to encrypt the message (i.e., in decimal digit) with the help of receiver's public key, i.e., 'e' and 'n'.
In decryption process, the receiver has to decrypt the ciphertext with the help of his private key, i.e., 'd' and 'n'.

6. CONCLUSION

When storing data on remote cloud storages, users want to be assured that their outsourced data are maintained accurately in the remote storage without being corrupted. In addition, cloud servers want to use their storage more efficiently. To satisfy both the requirements, we proposed a scheme to achieve both secure deduplication and integrity auditing in a cloud environment. To prevent leakage of important information about user data, the proposed scheme supports a client side deduplication of encrypted data, while simultaneously supporting public auditing of encrypted data. We used BLS signature based homomorphic linear authenticator to compute authentication tags for the PoW and integrity auditing. The proposed scheme satisfied the security objectives, and improved the problems of the existing schemes. In addition, it provides better efficiency than the existing schemes in the viewpoint of client-side computational overhead. Finally, we designed two variations for higher security and better performance. The first variance guarantees higher security in the sense that a legitimate user can be an adversary. The second variance provides better performance from the perspective of the clients, by permitting low-powered clients to perform upload procedure very efficiently.

REFERENCES

1. Efficient Client-Side Deduplication of Encrypted data with Public Auditing in Cloud Storage Taek-young youn1, ku-young chang1, kyung hyune rhee2, and sang uk shin2, (member, ieee) 1Electronics and Telecommunications Research Institute (ETRI), Republic of KOREA Dept. of IT Convergence and Application Engineering, Pukyong National University, Republic of KOREA
2. Secure Data Deduplication in Cloud 1Arpitha. R, 2Pavithra G. S M. Tech Student, Associate Professor 12Department of CSE, SJBIT, Bangalore
3. DeDu: Building a Deduplication Storage System over Cloud Computing Zhe SUN, Jun SHEN School of Information Systems and Technology Faculty of Informatics, University of Wollongong, Wollongong, NSW, Australia; Jianming YONG School of Information Systems Faculty of Business, University of Southern Queensland, Toowoomba, QLD, Australia
4. Improved Proxy-Encryption Schemes with Applications to Secure Distributed Storage, Giuseppe Ateniese, Kevin Fu, Matthew Green, Susan Hohenberger
5. Encrypted Big Data Zheng Yan, Senior Member, IEEE, Wenxiu Ding, Xixun Yu, Haiqi Zhu, and Robert H. Deng, Fellow, IEEE
6. Flexible Data Access Control Based on Trust and Representation in Cloud Computing Zheng Yan, Senior Member, IEEE, Xueyun Li, Mingjun Wang and Athanasios V. Vasilakos, Senior Member, IEEE
7. ClouDedup: Secure Deduplication with Encrypted Data for Cloud Storage Pasquale Puzio, SecludIT and EURECOM; Refik Molva, EURECOM; Melek O¨ nen, EURECOM; Sergio Loureiro, SecludIT; Sophia-Antipolis, France