Cloud Storage: A Review on Secure Deduplication and Issues

S. Annie Joice, M. A. Maluk Mohamed

Computer Science and Engineering, Anna University, India
anniejoice@gces.edu.in, malukmd@mamce.org

Abstract

Data deduplication in cloud is gaining popularity among cloud users because it enables cloud users to reduce the storage costs and the network bandwidth costs. Many security and privacy issues exist in general deduplication techniques and various secure deduplication techniques have been proposed to keep the sensitive data secure. A diverse range of solutions has been proposed for secure deduplication, ownership challenge and deduplication in the cloud environment. In this article, deduplication systems are classified based on message dependent encryption, ownership and cloud architecture. Based on the classification, security risks and side channel attacks from inside and outside adversaries and potential problems in deduplication are explored. Each scheme is compared in terms of their security and efficiency. Finally, the challenges in existing deduplication systems in the cloud and future research directions and challenges are discussed.

Keywords: Cloud, Convergent encryption, Deduplication, Message dependent encryption

1 Introduction

It has been predicted that by 2020 Global Data Center traffic to reach 15.3 Zettabytes annually. A maximum workload percentage of up to 92 is being processed in Cloud. Storing data in Cloud is particularly high demand and will cost high in future. Due to the rapid growth of data, multiple users may store replica of the same data within cloud storage.

Storage is a service model in cloud in which data can be stored, managed, and made available to users over Internet. The operating expenses incurred by Cloud Storage providers are very high.

In public cloud infrastructure, redundant copies of the same file exist in cloud storage. Deduplication is a technique which eliminates redundant chunks of same data in the storage. It eliminates redundant files in the storage and keeps only a single copy of the file. By performing deduplication, storage costs can be reduced in standard file systems by more than 50% systems and for backup applications, it can be reduced up to 90% to 95% [1].

In addition to the operating expenses, security is an important concern among users. Secure Deduplication is an important requirement for cloud storage services. Preserving data security while performing deduplication is a challenging task. The major goal of secure deduplication is to provide both space efficiency and to protect data from adversaries.

1.1 Scope and Contribution

The first contribution of this article is to identify the evaluation criteria for secure deduplication systems: deduplication overhead, security, scalability, and reliability. The second contribution is identifying the key design decisions: data granularity, deduplication location, indexing, and deduplication technique. Based on the design decision different methodologies used for each of them are explored. In this survey security and efficiency of the various existing systems and their limitations are analyzed thoroughly. Finally, the security issues in present deduplication systems are identified and design decisions for deduplication in Mobile Cloud Computing (MCC) environment is discussed.

1.2 Related Work

Many extensive surveys on general deduplication techniques are performed. Mandagere et al. [2] characterized the taxonomy for existing deduplication systems in three key dimensions: Placement, timing, and algorithm. Experimental evaluations were conducted by applying different deduplication algorithms on the backup data set. Resource utilization such as CPU utilization, deduplication time and CPU cycles based on different techniques and their performance is evaluated.

Meyer and Bolosky [1] presented a survey on practical deduplication. In their work, the relative efficiency of the system in file-level deduplication and block-level deduplication is analyzed thoroughly on Windows File Systems. The survey is performed on diverse file systems and various file size.

Paulo and Pereira [3] explored the existing deduplication systems and presented the taxonomy based on design decisions. Deduplication on storage

*Corresponding Author: S. Annie Joice; E-mail: anniejoice@gces.edu.in
DOI: 10.3966/160792642019052003019

systems are analyzed thoroughly and explored based on their performance measures.

Shin et al. [4] presented an extensive survey on deduplication in cloud storage systems. Security on existing deduplication systems based on various threats was analyzed. Deduplication based on encryption techniques, Proof of Ownership (PoW) was thoroughly analyzed in terms of their performance and security and their advantages and disadvantages are discussed.

1.3 Organization of the Paper

The rest of the paper is organized as follows. Section 2 presents the parameters used to evaluate different deduplication schemes. In Section 3 design decisions for secure deduplication. Section 4 discusses the security threats to secure deduplication systems. Section 5 presents the taxonomy of secure deduplication. Section 6 deals with the actual survey of different deduplication schemes that have been presented and published. The security features and the comparison of various security schemes are analyzed and discussed in section 7. Future research directions and challenges are discussed in Section 8. Finally, Section 9 concludes our survey.

2 Evaluation Criteria for Deduplication

In this survey, a number of security frameworks are presented that deals with secure deduplication and secure key management. The security frameworks are evaluated based on their performance, and security features.

2.1 Duplication Overhead

Deduplication overhead comprises computation overhead, storage overhead and communication overhead involved in the system.

2.2 Scalability

The ability of the deduplication system to work with the increased file size and increased demand.

2.3 Reliability

The ability of the system to be consistent with the repeated deduplication operation and helps to avoid data loss.

3 Design Decision Criteria

3.1 Data Granularity

Based on the minimal data size checked by the system for redundancy, deduplication can be performed in various ways. In File-Level deduplication, a unique identifier called hash number or hash signature is generated for the entire file using a hash algorithm. It is stored as an index for the file and used to find duplicates stored in the cloud. File-level deduplication can be performed easily. Processing overhead is very less because the hash of the file can be easily generated and less overhead in maintaining the metadata of the file [1, 5].

In block-level deduplication, the file is divided into multiple blocks of fixed or variable length. Deduplication is performed at block-level. For variable length blocks, Rabin fingerprinting scheme [6] is applied to generate hash signatures. To chunk a file, start from the beginning of the file and looks for the byte stream to meet certain criteria, which defines the boundary of the chunk. Variable length blocks are generated using sliding window mechanism [7]. Then the cryptographic hash for the chunk is calculated. Block-level deduplication requires more processing overhead than the file-level deduplication since the number of chunks needs to be processed is high [8]. Tracking the index of the blocks in each iteration also gets larger. Sometimes in variable length blocks, the same hash number may be generated for two different blocks, which may lead to hash collisions. At that time, the storage will not save the new blocks, as the hash number already exists in the index file.

To create large chunks of data, a dynamic partitioning algorithm Fingerdiff [7] is used. In this algorithm, the chunk size is reduced in the regions expected to change and in the regions unaffected by changes, the chunk size is kept large.

3.2 Deduplication Location

Deduplication can be performed at various location. Source-based, target based, and in-line. In source-based deduplication, redundant blocks or files are removed before transmitting to the target storage (cloud storage). It reduces bandwidth usage. For target-based deduplication, blocks of data or files are transmitted across the network to target storage. This method is preferable for large volumes of dataset. In-line deduplication eliminates redundant data before it is being stored in the target storage. If the target storage identifies the file or block of data already stored, an index to the existing block is stored, rather than the whole block or file.

3.3 Indexing

To find and removing duplicates is a resource-intensive task, so suitable data structure for indexing is important. Indexing can be performed by computing the hash signature of the file, which can be used to find duplicates. Generating hash signature requires additional processing overhead which is not suitable for resource-constrained devices [9]. In private deduplication scheme [10], the cloud server stores only small information about the file to improve the performance. The server can verify for duplicates without fetching the entire file. For variable-sized

chunks, common fingerprints are compared by computing a set of Rabin fingerprints. To avoid number of comparison, similar fingerprints are grouped together into superfingerprints [7]. For a superfingerprint with high resemblance, the index is scaled to a larger number of chunks.

3.4 Cloud Architecture

Deduplication can be performed on various cloud architecture such as (i) Single cloud (ii) multicloud and (iii) hybrid cloud. In single cloud architecture, convergent keys and Proof of Ownership (PoW) mechanisms can be employed to protect the data from data loss and data breaches. This is the most common approach followed by many of the commercial CSPs. To avoid the single point of failure in single cloud architecture, multicloud architecture divides the file into multiple shares and it is distributed across multiple cloud server. To avoid disaster recovery [11] in multicloud, optimized scheduling strategies can be applied to achieve data reliability and short recovery time.

In hybrid cloud architecture [12], authorized data deduplication is performed. In this method outsourced data is stored in public cloud and all data management operations is handled in private cloud. The user can perform duplicate check, if the user meet the specific privileges. Secure deduplication can be done by encrypting the user file with different privilege keys.

4 Security Threats in Cloud Deduplication

There exist enormous challenges in the cloud including trust, security, and privacy of the outsourced data. The data owner outsources the data to CSP, which in turn lead to security risks regarding the privacy of the outsourced data. Deduplication techniques performed in cloud storage has security risks and the possibility of revealing information about the contents of the file stored in cloud storage. The security issues include the security and privacy of the data stored in the cloud, security threats from inside and outside adversaries.

A secure communication channel is needed between the cloud and the user due to the establishment of the covert channel by adversaries. This can be done by securing the communication channel by routing protocols.

4.1 Side Channel Attacks

In deduplication, attacks can occur either at the file-level or block-level. During the working of a cryptosystem, some physical activities can reveal useful information about secrets in the system. The degradation of secret information results in side-channel leakage [13]. Source-side cross-user deduplication [14-15] can be used by an attacker to learn sensitive information about the user. The degradation of secret information results in side-channel leakage.

An attacker can perform two types of attacks on online storage services:

4.1.1 Learning the Contents of Other User Files

If an attacker suspects the existence of sensitive information in cloud storage, the attacker can perform deduplication to check whether the same copy exists or not. If deduplication occurs, an identical copy exists in the storage. To learn the contents of the file, an attacker can perform this attack over all possible range of values in the file contents. If deduplication occurs on a single copy of the file, the attacker can able to know the file.

4.1.2 Establishing a Covert Channel

Deduplication can also be used to establish covert channel from the user system to remote cloud storage through the software that runs between the system and cloud storage.

5 Taxonomy of Secure Deduplication

An important goal of secure deduplication scheme is to provide the solution to various security threats and performs deduplication on cloud storage securely and efficiently. Based on the design decision criteria discussed above, deduplication schemes can be categorized into 3 different approaches: Message-dependent encryption, PoW, Cloud Architecture. The taxonomy of secure deduplication is shown in Figure 1.

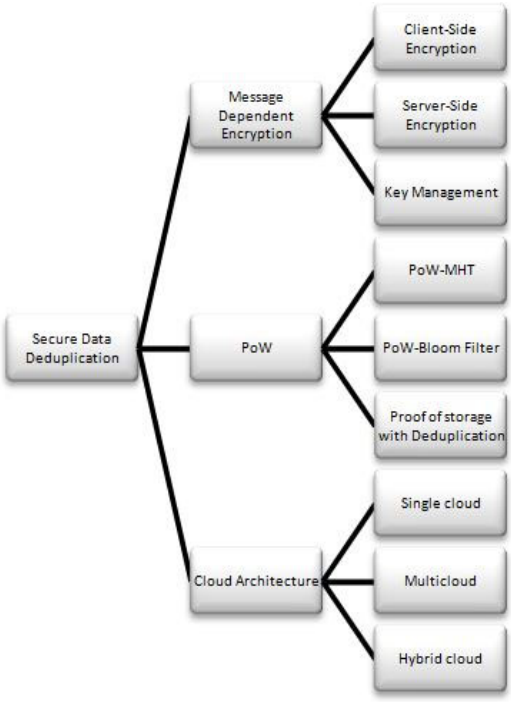

Figure 1. Taxonomy of secure deduplication

5.1 Message-Dependent Encryption

Encryption techniques are applied to outsourced data to protect the confidentiality and privacy of data stored in the cloud. Key generation and management play a vital role in encryption algorithms. In message-dependent encryption, the encryption key is generated from the message itself. It can be performed either on client-side or server-side based on the location encryption is performed. Convergent threshold encryption [16] is a combination of convergent encryption and threshold encryption scheme. In this method, sensitive data shared by multiple data owners, encrypted by different encryption keys are converted into a single convergent encryption ciphertext. If the number of duplicates reaches the predefined threshold value, CSP converts the outsourced encrypted data [17] into a single convergent encryption ciphertext.

5.2 Proof of Ownership (PoW)

It is necessary for the client, to prove the ownership of data to the cloud server. There is a possibility of leakage of the hash value of the file and it can be used by malicious users. PoW addresses the problem of ownership, unauthorized access by malicious users and other side-channel attacks. PoW based solutions [18] are classified as: Merkle hash tree based solution, (PoW-MHT), Bloom filter, Proof of storage with Deduplication.

5.3 Cloud Architecture

Based upon the cloud architecture, deduplication can be performed either on client-side or server-side. In single cloud architecture, deduplication can be performed either on client-side or server-side. In multicloud architecture, deduplication is performed across multiple cloud servers. In hybrid cloud architecture, the public cloud is used to store the outsourced data and private cloud is used to perform data management operations.

6 Survey of Existing Deduplication Schemes

6.1 Secure Deduplication with Efficient and Reliable Convergent Key Management

Li et al. [19] proposed two schemes for key management in secure deduplication. Convergent encryption [20] provides data confidentiality while performing deduplication. The convergent key is obtained by computing the hash value of the data to be stored in the cloud. By using the convergent key generated, data encryption is performed, cipher text is stored in the cloud, and the user holds the key. Since this encryption technique is deterministic, identical copies of the same data will generate the same convergent key, which in turn generates the same cipher text.

6.1.1 Baseline Approach

This scheme involves the user and the Storage Cloud Service Provider (S-CSP). In this approach, the convergent key generated by the user is then encrypted by an independent master key. The user holds the master key, while the convergent keys are stored by S-CSP. This approach consists of:

Symmetric Encryption (SE) scheme with the following primitive functions: $KEYGEN_{SE}$, $ENCRYPT_{SE}$, $DECRYPT_{SE}$, and the users master key is initialized as $k=KEYGEN_{SE}(1^{\lambda})$, where 1^{λ} is some security parameter.

(1) A Convergent Encryption (CE) scheme consists of the following primitive functions: $KEYGEN_{CE}$, $ENCRYPT_{CE}$, $DECRYPT_{CE}$, $TagGen_{CE}$.

(2) Proof of Ownership (PoW) algorithm for the file (PoW_F) and for the block (PoW_B).

In this approach, the S-CSP is initialized with two types of storage systems: a rapid storage system to store the tags that performs duplicate checks, and a file storage system to store both encrypted data copies and convergent keys.

The user computes the file tag $T(F)= TagGen_{CE}(F)$ and sends it to the S-CSP. On receiving the tag file T(F), the S-CSP checks whether the same tag exists on the S-CSP. If the same tag exists, then the S-CSP replies user with “file duplicate” response or “ no file duplicate response” otherwise.

6.1.2 Dekey Approach

Dekey approach reduces the storage overhead on key management compared to Baseline approach. The Dekey approach also solves the problem of single point of failure on master key. Instead of performing encryption on convergent keys for an individual user, Dekey approach constructs secret shares on the plain convergent keys and distributes the secret shares across multiple Key Management Cloud Service Providers (KM-CSP). In this approach, file-level deduplication is same as that of baseline approach. The next stage after file-level and block-level duplicate checks is key distribution.

If the response is “file duplicate” from S-CSP, to prove the ownership $PoW_{F,j}$ is performed for the tag $T_j(F)=TagGen_{CE}(F,j)$ with the *j-th* KM-CSP. If PoW is passed, all the pointers for the secret shares of F will be sent to the user.

If the response from S-CSP is “no file duplicate”, the following operations are performed:

For each block B_i, tag block $T(B_i)=TagGenCE(B_i)$ is computed by the user and send to each KM-CSP. In addition to that, a file tag $T_j(F)=TagGen_{CE}(F,j)$ is also computed and sent to the *j-th* KM-CSP, $1 \leq j \leq n$.

Upon receiving the tag, $POW_{B,j}$ is performed for the

block. If POW_B is passed, *j-th* KM-CSP will send the secret share corresponding to the convergent key K_i to the user. If POW_B is failed, KM-CSP sends a signal to the user to send secret share on convergent key. Then the user computes the secret share using (n,k,r) Ramp Secret Sharing Scheme (RSSS) to generate shares K_{i1}, K_{i2},…, K_{ik}. Further, it sends the share K_{ij} and tag to the *j-th* KM-CSP for *j=1,2,…n*. KM-CSP stores the share and tag for the block and returns the pointer to the user for future access.

In this scheme, downloading a file from the cloud is identical to baseline approach. The user fetches the corresponding secret shares K_{ij} for each block B_i and reconstructs convergent key K_i for B_i. Finally the downloaded blocks C_i can be decrypted with $\{K_i\}$.

The author further evaluated the encoding and decoding performance of Dekey approach to generate and to recover shares respectively. The author concluded that encoding/decoding overhead in Dekey approach is less compared to network transmission overhead in file upload/download.

6.2 Deduplication on Encrypted Big Data in Cloud

Zheng Yan et al. [21] proposed a deduplication scheme for encrypted data stored on the cloud based on ownership challenge and proxy re-encryption integrated with access control. The system model contains three entities (a)Cloud Service Provider (CSP) (b) data holder and (c) authorized party (AP). A procedure for Deduplication scheme is shown in Figure 2.

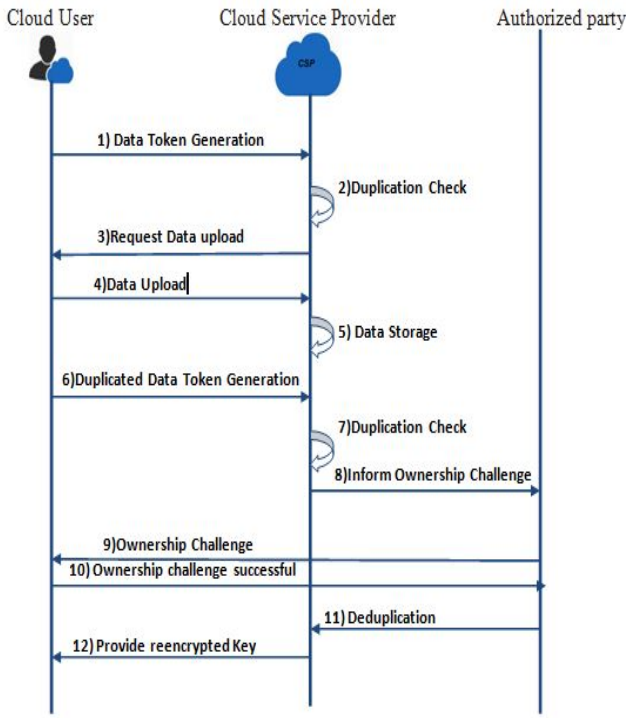

Figure 2. A procedure for deduplication scheme

The authors have discussed Encrypted Data Upload, Data Deduplication, Data Deletion, Data Owner management, Encrypted data update. To achieve secure data deduplication Elliptic Curve Cryptography (ECC), Proxy Re-encryption (PRE), and symmetric encryption is applied. A PRE is a polynomial time algorithm with 5 tuples (KG; RG; E; R; D): where KG, E, and D are the standard key generation, encryption and decryption algorithms. The re-encryption key generation algorithm of PRE RG takes private and the public key pair (pk_A, sk_A, pk_B) and generates re-encryption key for proxy $rk_{A\rightarrow B}$. The re-encryption algorithm of PRE R generates C_B which can be decrypted with the private key sk_B. $R(rk_{A\rightarrow B};C_A)=E(pk_B;m)=C_B$

The PRE is based on bilinear mapping $e:G_1 X G_1 \rightarrow G_T$, where G_1 and G_T of prime order q. Every data holder in the system setup generates secret key sk_i and public key pk_i for PRE. $sk_i = a_i$, $pk_i = g^{a_i}$ where $a_i \in Z_p$. To verify the unique identity of the user u_i the keys (pk_i ; sk_i) and (V_i;s_i) where $s_i \in R\{0,....2^{\sigma}-1\}$ is the ECC secret key of the user u_i over the finite field $GF(q)$ and $V_i=-s_i P$ is the corresponding public key and σ is the security parameter. AP independently generates public key pk_{AP} and secret key sk_{AP} for PRE and broadcast the public key pk_{AP} to the users of CSP.

The proposed scheme consists of the following phases:

(1)Encrypted Data Upload: User u_1 generates data token for the sensitive data M $x_1=H(H(M)X P)$and sends (x_1, pk_1,cert(pk_1)) to CSP.

(2) Data Deduplication: CSP checks whether the duplicated data exists by verifying cert(pk_1) such that x_1 exists or not. If x_1 does not exist user u_1 encrypts data M with the symmetric key of u_1 DEK_1 to get ciphertext CT_1. To obtain cipher key encrypt DEK_1 with pk_{AP} to get cipher key CK_1. User u_1 sends (CT_1, CK_1) to CSP which is saved along with x_1 and pk_1. CSP informs the user if the duplication check is positive and from the same user. If the data is same and from the different user deduplication is performed.

6.2.1 Data Deletion

If data holder u_2 wants to delete the data, the user sends deletion request to CSP $cert(pk_2), x_2$. CSP first verifies the validity of the request by the user u_2, and then deletes the duplication record and blocks later access by the user. If the deduplication record is empty, the CSP deletes encrypted data CT and deleted records.

6.2.2 Data Owner Management

If the data owner u_1 uploads the data after the data holder u_2, the data owner should prove the ownership, by providing the certificate. To know the corresponding re-encryption key of all data holders *i*, CSP sends the request to AP by providing their public key pk_i. If the ownership challenge is positive, AP issues $rk_{AP}\rightarrow u_i$ ($rk_{AP}\rightarrow u_2$) to CSP. CSP re-encrypts

CK_1 with $rk_{AP} \rightarrow u_2$ and get $E(pk_2, DEK_1)$ and remove CT_1 and CK_1 of user u_1. Finally, the corresponding deduplication records are updated.

6.2.3 Encrypted Data Update

If a user u_1 wants to update data, encrypt data M with DEK_1' to get CT_1'. Encrypt DEK_1' with pk_{AP} to get CK_1'. User u_1 send an update request: {x_1, CT_1', CK_1', update CT_1'} to CSP, CT_1', CK_1' together with x_1 and pk_2 by CSP. If the re-encryption key for other data holders is not known, CSP sends the request to AP for deduplication for other data holders. AP checks its policy and generate $rk_{AP} \rightarrow u_2$ and send it to CSP. CSP re-encrypts CK_1' with the re-encryption key to get E(pk_2, DEK_1') and remove CT_1 and CK_1. The re-encrypted keys are sent to the all eligible data holders for encrypted data update and future data access on M.

The authors perform security analysis and performance evaluation on the proposed scheme and concluded that the scheme reduces the storage space of CSP and efficiently perform big data deduplication.

6.3 Message-Locked Encryption and Secure Deduplication

Message-Locked Encryption [22] (MLE) is a new cryptographic primitive in which the key derived from the message itself is used to perform encryption and decryption. Bellare et.al introduced MLE and demonstrated its practical and theoretical contributions. In MLE Scheme:

i. k: Message M is mapped to Key K.

ii. Encryption Algorithm ε : Cipher text C is produced from message M using key K.

iii. Decryption Algorithm D: M is recovered from C using key K.

iv. Tagging Algorithm T: Cipher text C is mapped to a tag T, which is used to detect duplicates by the server.

Convergent encryption (CE) is viewed as MLE scheme that lets K=H(M), C=E(K,M), and T=H(C). Numerous variants of message-dependent encryption scheme such as CE, HCE1, HCE2, and RCE are proposed and these schemes are analyzed in terms of security properties and tag consistency. Tag Consistency (TC) is used to make integrity violations impossible and it is achieved by comparing the tag match $T(C_A)=T(C_B)$. In Strong Tag Consistency (STC) decryption of outsourced cipher text is different from M.

In symmetric key encryption SE, Concatenation ||, with STC CE performs K=H(M), C=SE(K,M), D=SD(K,SE(K,M)) and tag T=H(C). For HCE1, K=H(M), C=SE(K,M)||H(K), D=SD(SE(K,M)) and T=H(K) without TC.

HCE2 and RCE are the other two new schemes. HCE2 is identical in efficiency as HCE1. RCE is more efficient, compared to other schemes. In RCE, encryption is performed by picking up a random key and then generating an appropriate tag for ciphertext in the same pass.

In theoretical contribution, MLE deduplication scheme cannot achieve semantic security. In MLE, the key is generated from the message itself, so it is possible for an adversary to gain partial information on the message. But semantic security can be achieved using MLE, given unpredictable messages.

The four MLE approaches achieve privacy against chosen distribution attack (PRV-CDA). MLE can be performed either on client-side or server-side.

6.4 Interactive Message-locked encryption and Secure Deduplication

Bellare and Keelveedhi extended their prior work MLE to interactive Message-Locked Encryption [23] (iMLE) in which interactive protocols are used between client and server for upload and download operations. In iMLE, incremental updates can be performed using update protocol.

6.5 DupLESS: Server-Aided Encryption for Deduplicated Storage

Bellare et al. proposed secure server-side deduplication technique for encrypted data DupLESS [24] (Duplicate less encryption for simple storage) that provides security against brute force attacks. In this approach key server (KS) is used to generate keys instead of generating by the hash of messages.

DupLESS uses an obvious PRF (OPRF) protocol to obtain the key derived from message between KS and clients. The client uses the hash function H: $\{0,1\}^* \rightarrow Z_N$, RSA exponent e, and RSA modulus N which is used to compute blinded hash of the message m $x \leftarrow H(M)$ re mod N and is sent to the KS. The KS computes $y \leftarrow x^d \bmod N$, $ed \equiv 1 \bmod \varphi(N)$ and sends the result y back to the client. The client then removes the blinding signature and computes $z \leftarrow y \bullet r^{-1} \bmod N$. The result is computed as G(z) if and only if $Z^e \bmod N \neq H(M)$, $G: Z_N \rightarrow \{0,1\}k$ is a hash function. In DupLESS scheme, KS is not aware of client input and resulting PRF output and in the same way, clients are not knowing about the key.

The authors measure the performance of DupLESS scheme and concluded the bandwidth overhead diminishes with large file size and storage overhead is high. The security level provided by DupLESS is better than CE scheme.

6.6 Proof of Ownership in Deduplicated Cloud Storage with Mobile Device Efficiency

The author proposed PoW scheme [25] that provides balanced server side and user side efficiency. An illustrative example of the proposed PoW framework is shown in Figure 3.The Cloud storage constructs an

empty bloom filter during initialization. The cloud partitions the first copy of the file f into fixed length blocks l. The elements $h(f_i||f_n)$, $1 \le i \le (|f|/\lambda)$, where the *i-th* block of f_i is inserted into B. Then $< h(f), (|f|/\lambda) >$ is kept in memory and file f is stored in the disk. If the cloud server receives a request for the file, it checks whether h(f) is already in memory or not. If it does not exist, it ignores the request. Then the cloud server randomly chooses q distinct numbers from [1, $|f|/\lambda$]. The user is asked to reply with h (f_{di}|| f_n) and confirms the ownership of f.

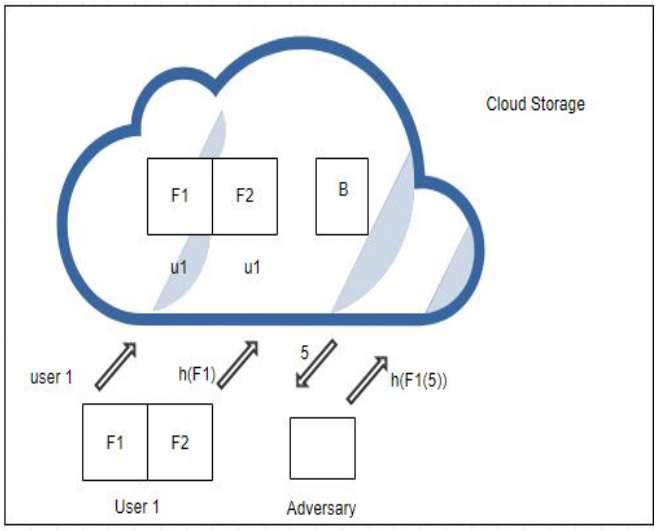

Figure 3. Proposed PoW framework

Dynamic bloom filter is used to keep the memory usage minimum and false positive probability low. An important feature of the dynamic bloom filter is its extensibility to hold more elements. By this approach, disk access overhead in server side is reduced, but a single bloom filter is to be stored in memory for each file.

In this scheme, the I/O latency is reduced in both server side and user side using bloom filter.

6.7 Proof of Ownership (PoW) in Remote Storage Systems

Halevi et al. [26] proposed a Merkle Tree based PoW solution [27]. Merkle Hash Tree (MHT) is a tree in which every leaf node holds data blocks and every non-leaf node holds the cryptographic hash of its child nodes. In this method, erasure coding is applied to the file content.Let $E:\{0,1\}^M \rightarrow \{0,1\}^{M'}$ be an α-erasure code, where α>0 and H a collision-resistant hash function. The Merkle binary tree over buffer X using leaves of b-bit and hash function H is denoted as $MT_{H,b}(X)$. For the input file $F \in \{0,1\}^M$ (M-bit input File), the verifier(Cloud Service Provider, CSP) computes X=E(F) and then construct Merkle tree $MT_{H,b}(X)$. After verification, it keeps only the root and the number of leaves in MHT. At the time of initiation of proof protocol, the verifier chooses at random, leaf indices $l_1 \ldots\ldots l_u$ where u is the smallest integer $(1-\alpha)u<\epsilon$. The leaf indices are sent to the user (client) to prove their ownership. The user returns the sibling path of all the leaf nodes to the root. The verifier verifies the response with respect to $MT_{H,b}(X)$ and returns "Accept" to the user. If not, it sends "Fail" response. In this approach, the user can prove the verifier, without sending the file. In MHT based PoW scheme, the data owner has to perform a number of computations and I/O operations to prove the ownership of the data file. Spot checking-based PoW solution proposed an enhanced PoW protocol compared to MHT based PoW scheme.

6.8 Secure and Efficient Proof of Storage with Deduplication

In cloud storage systems in addition to security, data integrity is also an important concern. In cloud storage security, two important notions are Proof of Data Possession [28] (PDP) and proof of Retrievability [29] (PoR). To verify the integrity of outsourced data in cloud PDP is used. PoR is used to recover the outsourced data from the cloud. Public Verifiability [30] can be achieved in both the aforementioned schemes, as they can verify the integrity of the file stored in the cloud. Zheng et al. proposed a proof of storage with Deduplication [30] (PoSD) which provides both data integrity and data deduplication. In this scheme, cryptographic key pairs are generated for integrity and deduplication and the tag is computed for data blocks by using the key pair. The file and its authentication tag are outsourced to the cloud storage and the server sets the authentication tag for integrity and deduplication as identical tags. To verify the integrity of the file stored in the cloud, the client sends the public key for integrity check and the file identifier. The cloud server performs the integrity check and if both hold return 1, else return 0.

6.9 Side-channels in Deduplication: Trade-offs between Leakage and Efficiency

In this paper [31], deduplication strategies to optimize efficiency and security are discussed. Due to the existence-of-file-attack, a deduplication strategy that uses file upload threshold which is based on some probability distribution is used. In this method, an adversary *A* attempts to simulate initial storage of file F by incrementing the counter *ctr* by one, as a result, the *store() oracle* return appropriate signal *sig*. The counter ctr is used to keep track of upload requests made for file F. If the number of queries by the adversary to the *store() oracle* exceeds B, the adversary *A* will always receive *sig=0* and thus unable to gain any information.

This paper concludes modeling attacks and analyzed solutions for file upload based on probabilistic distribution and strategies to defend against these

attacks.The upload request for the file at first is with the signal *sig*=1. A deduplication strategy that uses upload threshold value based on some probability distribution p_i probability for the threshold value is *i*. The bandwidth cost is measured in terms of number of expected uploads of each file $E = \sum_{i=1}^{\infty}(ip_i)$.

6.10 Improving the Resistance to Side-Channel Attacks on Cloud Storage Services

A new deduplication model gateway-based deduplication [32] is proposed to reduce the risk of information leakage. Intra-account deduplication is to remove redundant data from a single user account on the cloud. Inter-account deduplication is to remove redundant data from the whole set of users in the cloud. In this paper, the author proposed a client-based approach for inter-account deduplication. The proposed system is composed of five modules running at a different location: Cloud Storage Service Provider (SSP) server running at SSP premises, Gateway Server, Gateway Client, and Bandwidth Manager running on Network Service Provider (NSP) gateway at the customer premises. The Gateway based deduplication procedure is shown in Figure 4.

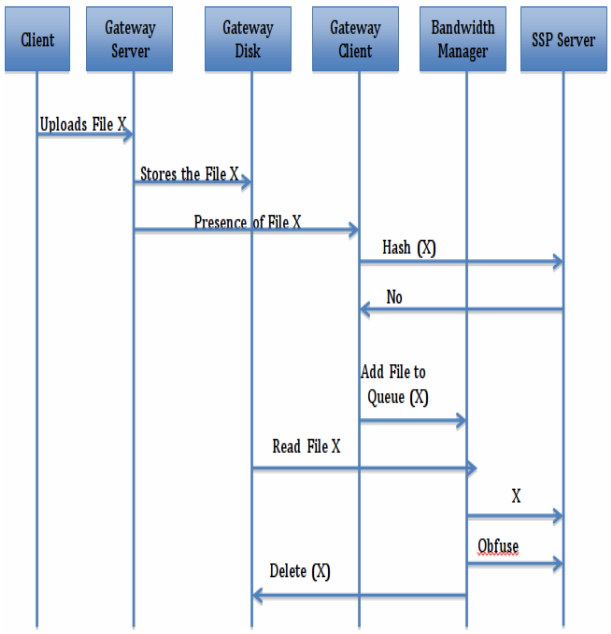

Figure 4. Gateway based deduplication procedure

The proposed system achieves significant bandwidth savings on the network with the help of SSP server and reduces side-channel attacks by the adversary. The security mechanism is employed between gateway and SSP to reduce the risk of side-channel attacks towards an adversary. The trade-off between cloud storage resistance to side channel attacks and savings in bandwidth is represented by a parameter α. Bandwidth savings are maximum if the value of α is 0. The bandwidth allocation for cloud storage is calculated as $B_c = \alpha * (B_{max}-B_T)$, where B_{max} is the maximum bandwidth and B_T is the bandwidth required for various cloud services.

6.11 Differentially Private Client-side Data Deduplication Protocol for Cloud Storage Services

The proposed system consists of three entities Cloud Storage Server (CSS), Storage Gateway (GW) and Users (U). The deduplication protocol is implemented on GW [33] to hide the deduplication process from adversary A. GW performs deduplication and reduces the volume of data to be stored on CSS. To improve the performance of the system, for file downloading, GW searches for F in the local storage and if found return the file F to U immediately. Otherwise, GW retrieves F from CSS.

The security analysis of the proposed scheme reduces the bandwidth consumption compared to the other randomized schemes [14, 34]. The proposed system improves network efficiency by utilizing the storage space of GW and reduces the risk of information leakage towards an adversary.

6.12 Privacy Aware Data Deduplication for Side-channel in Cloud Storage

In this paper, Zero-Knowledge deduplication response (ZEUS and ZEUS+), a privacy-aware deduplication protocol is proposed. By using the proposed system, two-side privacy is obtained with reduced cloud storage. No additional hardware is required. The proposed system incur slightly increased communication overhead.The file to be uploaded is divided into chunks and based on the dc response the chunks are uploaded. The communication cost to upload two chunks $C_1 \oplus C_2$ of length ϕ in ZEUS to achieve privacy is $p^2\phi$. ZEUS has less communication cost compared to ZEUS+ because the communication cost incurred by random threshold (RT) is added to ZEUS+.

6.13 Modeling the Side-Channel Attacks in Data Deduplication with Game Theory

In this paper, a game-theoretic approach [35] is used to model the interaction between attacker and cloud service provider.The solution of the game-theoretic model is based on mixed strategy Nash equilibrium. The payoff matrices are defined for the attacker and the service provider.The proposed scheme provides an optimal decision for the cloud service provider by using payoff defined by the utility function compared to the threshold-based scheme. The number of uploaded copies of files depends on several factors such as payoff matrices, service provider gain from defending the game and convergence condition.

6.14 RARE: Defeating Side Channels Based on Data Deduplication in Cloud Storage

In this paper, RAndom REsponse (RARE) approach [36] is used to eliminate deduplication. Duplicate Check is performed on two chunks at once. To upload a file F on cloud S, F is divided into chunks C. To upload a chunk C, deduplication check request h(C) is uploaded. In double chunk uploading for the dc request $<h(C_1), h(C_2)>$, the dc response represents a single value that indicates the total number of chunks to be uploaded. The RARE table is presented in Table 1.

Table 1. RARE

C_1 Existence on S	C_2 Existence on S	dc response from s
0	0	2
0	1	1 or 2
1	0	1 or 2
1	1	1 or 2

If both the chunks C_1 and C_2 are not in S, two individual chunks are required to upload or exclusive-OR (XOR) $C_1 \oplus C_2$ is uploaded. By using this design S can able to derive another chunk, given a chunk in existence.

Dirty bit is used to mark the chunks that have been queried for existence but are not uploaded on S. if the dc response status is 2, it indicates that at least one queried chunk is available in S. A dirty chunk list is maintained and in duplicate check if $<h(C_1), h(C_2)>$ is in the dirty chunk list, S always return response as 2.

The proposed scheme RARE achieves side-channel defense with the weak existence privacy and inexistence privacy. The RARE table along with dirty chunk list ensures the privacy of chunk existence status on S. The communication cost includes the total number of bits required for uploading chunk (bc), including duplicate chunk(dc) and explicit chunk uploading(ec).

7 Security Analysis and Comparison

In this section security analysis of various deduplication schemes are evaluated based on the criteria presented in Table 2.

Table 2. Security analysis of various deduplication schemes

Framework	Basic Theory	Confidentiality	Availability	Authenticity	Side Channel Resistance	Integrity	Security Feature	Limitations
(Liet al., 2014)	Convergent Encryption Baseline Approach	Yes	Yes	Yes	Unsatistactory	Yes	Convergent keys are encrypted by an independent Master Key	Storage overhead-Key Management, Unreliable
(Liet al., 2014)	Convergent Encryption Dekey Approach	Yes	Yes	Yes	Unsatistactory	Yes	Reliable Key Management	Bottleneck in encryption or decryption
(Yan, Ding, Yu, Zhu, & Dent, 2016)	Ownership Challenge and Proxyre-encryption	Yes	Yes	Yes	Unsatistactory	Yes	Data deduplication canbe performed-offline	Computation overhead
(Bellare, 2013)	Message Locked Encryption	Yes	No	No	Unsatistactory	Yes	Compatible with client/serverside deduplication	Abuse of services
(Bellare, & Keelveedhi, 2015)	Interactive Message Locked Encryption	Yes	No	No	Unsatistactory	Yes	Incremental updates	Computational overhead
(Bellare, Keelveedhi, 2015)	DupLESS Server-side deduplication	Yes	No	Yes	Unsatistactory	Yes	Resistance to external attacks	Single point of failure
(Yu, Chen, & Chao, 2015)	PoW-Bloom Filter	No	No	Yes	Unsatistactory	No	PoW-unauthorized file download	Overhead in index management
(Halevi, Harnik, Pinkas, & Shulman-Peleg, 2011)	PoW-MHU	No	No	Yes	Unsatistactory	No	Resistance to attacks on client side	Inefficient due to the use of erasure code
(Zheng & Xu, 2012)	PoSD	No	No	Yes	Unsatistactory	Yes	Simultaneously performs deduplication and proof of storage intergrity	Cannot deal dynamic data

The various deduplication schemes are compared based on the efficiency and the results are presented in Table 3, where n is the number of chunks produced from file f, s is the size of file blocks, λ is a security parameter, m is the number of data holders. The computation cost of various operations is represented as follows: Hash is a hash evaluation, Mul is the multiplication operation, Exp is the Exponentiation operation, Point Mul is the Point Multiplication operation, Sym.Enc is the Symmetric Encryption operation, Pair is the pairing operation, ModInv is the Modular Inversion operation. In iMLE the computation complexity of encryption and homomorphic encryption varies depending on the lattice, R-LWE(ring Learning with Errors) assumptions. So the computation overhead of the scheme is ignored in the efficiency comparison. In Ownership challenge and proxy re-encryption scheme, data token is stored along with the data which requires an additional 1184 bits.

The Client-side deduplication is generally preferred over Server-side deduplication in terms of bandwidth usage, communication cost, and storage cost. The average number of uploads of a file is measured in terms of the bandwidth usage and the comparison of communication costs is presented in Table 4.

Table 3. Efficiency comparison of various deduplication schemes

Framework	Basic Theory	Computation Overhead		Storage Overhead		Communication Overhead
		Client	Server	Client	Server	
(Li et al., 2014)	Convergent Encryption Dekey Approach	O(n) Hash+ O(n) Mul+ O(n) Exp+ O(n) Sym.Enc	O(n) Exp+ O(1) Pair	O(1)	O(1)	O(1)
(Yan, Ding, Yu Zhu, & Deng, 2016)	Ownership Challenge and Proxy re-encryption	O(1) Point Mul+ O(1) Exp (Data Owner) O(m) Exp+ O(m) Point Mul (Data holder)	O(n) Pair+ O(1) ModInv+ O(1) Exp	O(1)	O(1) +1184 bits (Token)	O(1) Exp+ O(1) Pair
(Bellare, 2013)	Message Locked Encryption	O(1) Hash+ O(1) Sym.Enc	-	O(1)	O(1)	O(1)
(Bellare, Keelveedhi, & Ristenpart, 2013)	DupLESS: Server-side deduplication	O(1) Hash+ O(1) Mul+ O(1) Exp+ O(1) sym.Enc	O(1) Exp	O(1)	O(1)	O(1)
(Yu, Chen, & Chao, 2015)	PoW-Bloom Filter	O(1) Hash	O(lk) Hash	O(f)	O(f)	O(f)
(Halevi, Harnik, Pinkas, & Shulman-Peleg, 2011)	PoW-MHT	O(n) Hash	O(log λ) Hash	O(1)	O(1)	O(log λ)
(Zheng & Xu, 2012)	PoSD	O($s\lambda$) Mul	O($s\lambda$) Mul+ O(λ) Exp	O(1)	O(m)	O(s)

Table 4. Comparison of communication cost (Resistance to side-channel attacks)

Framework	Basic theory	Communication cost
Heen et al. (2012)	Gateway Disk is used	$B_c = \alpha * (B_{\max} - B_T)$
Shin et al. (2015)	Storage Gatway is used	$B = 0$
Wang et al. (2015)	Game Theoritic Approach-Mixed Strategy Nash Equilbrium	Sensitive to several factors
Amknecht et al. (2017)	Based on Upload threshold	$B = \sum_{i=1}^{\infty} (ip_i)$
Yu et al. (2018)	Dirty chunks are used. No extra hardware.	$P^{2\phi}$
Zahra et al. (2018)	Dirty chunks are used.	$B = bc + dc + ec$

8 Discussion and Future Directions

In cloud deduplication systems, many problems in security, privacy, integrity, and reliability have solved, but still there exist open research challenges. The unsolved issues are discussed in this section.

8.1 Data Dependency and Privacy Issues

In convergent encryption, the convergent key is generated from the file itself, and the efficiency of this method is data dependent. Deduplication performed using convergent encryption leads to privacy issues. It can be used to find the users storing the file if the

attacker holds the copy of the file. Multiple users may possess ownership of the ciphertext stored in the cloud. Some users may request the CSP to revoke their ownership list for the file. The revoked users can be able to access the data stored in the cloud, as long as they hold the encryption key derived from the file. Thus proper ownership management and revocation is an important challenge for secure deduplication.

8.2 Achieving Secure Deduplication

In deduplication, when encryption is performed on the client side, privacy is preserved. When deduplication is performed using random key by multiple users, different cipher texts are produced from identical files. To avoid this problem, convergent encryption is used, in which the key is derived from the data itself [19]. To achieve semantically secure deduplication, the data owner encrypts the data with the randomly generated encryption key and the key is distributed to another user, who share the data. In some schemes [24] encryption is performed by obtaining keys from the key server. When the key server is corrupted with the cloud server, this scheme will not work.

8.3 PoW in Multicloud Architecture

The data owner loss the control over the data, when the data is stored in the cloud. Data deduplication is vulnerable to data loss and data breaches in the cloud because only one copy of the data is stored in the cloud. In deduplication systems, researchers have focused their work to perform deduplication of encrypted data in the multicloud environment by distributing the shares across multiple cloud servers. This type of deduplication can protect the data from inside adversaries and CSPs. PoW for the outsourced data in a multicloud environment is important if client-side deduplication is performed. The reliability of outsourced data in multicloud architecture is also an important focus of research. To achieve security and availability using client-side deduplication in multicloud storage is important.

8.4 Deduplication in Mobile Cloud Environment

With the advent of mobile devices, and smart phones use of mobile cloud computing is rapidly increasing nowadays. The use of the bloom filter in PoW scheme [25] for mobile device leads to storage problem and indexing problem. As the mobile devices are resource constrained devices, a more succinct data structure for indexing the file is an important research focus. In source deduplication, network bandwidth can be saved by avoiding transmission of duplicate files over the network. As deduplication process is a relatively slower process, improving the efficiency of deduplication as the amount of data grows is also important. New models for redundant data identification [37] and cleaning methods in Mobile cloud architecture is an important research focus.

9 Conclusion

Data deduplication is an effective technique to reduce the storage costs and to save the network bandwidth. The existing deduplication systems are classified based on the design decisions. Security analysis of existing deduplication systems in the cloud are analyzed and discussed. The efficiency and security of the existing schemes are compared based on the evaluation criteria. Further, it discusses the future research scope and challenges in secure deduplication systems. As the growth rate of data is increasing day-by-day, secure deduplication is an important area of research focus. Different secure deduplication schemes in the cloud are critically investigated in this survey article. To achieve secure deduplication, security threats need to be studied and solved accordingly.

References

[1] D. T. Meyer, W. J. Bolosky, A Study of Practical Deduplication, *ACM Transactions on Storage*, Vol. 7, No. 4, pp. 1-20, January, 2012.

[2] N. Mandagere, P. Zhou, M. A. Smith, S. Uttamchandani, Demystifying Data Deduplication, *ACM/FIP/USENIX International Middleware Conference*, Leuven, Belgium, 2008, pp. 12-17.

[3] J. Paulo, J. Pereira, A Survey and Classification of Storage Deduplication Systems, *ACM Computing Surveys*, Vol. 47, No. 1, pp. 1-30, July, 2014.

[4] Y. Shin, D. Koo, J. Hur, A Survey of Secure Data Deduplication Schemes for Cloud Storage Systems, *ACM Computing Surveys*, Vol. 49, No. 4, pp. 1-38, Februray, 2017.

[5] D. Harnik, O. Margalit, D. Naor, D. Sotnikov, & G. Vernik, Estimation of Deduplication Ratios in large Data Sets, *28th Symposium on Mass Storage Systems and Technologies*, Pacific Grove, CA, 2012, pp. 1-11.

[6] A. Z. Broder, Proceedings of the Compression and Complexity of SEQUENCES 1997, *IEEE computer Society*, 1997.

[7] A. Z. Broder, *Sequences II Methods in Communication, Security, and Computer Science*, Springer-Verlag, 1993.

[8] M. W. Storer, K. Greenan, D. D. E. Long, E. L. Miller, Secure Data Deduplication, *15th ACM Conference on Computer and Communications Security*, Alexandria, VA, 2008, pp. 1-10.

[9] F. Chen, T. Luo, X. Zhang, CAFTL: A Content-Aware Flash Translation Layer Enhancing the Lifespan of Flash Memory Based Solid State Drives, *9th USENIX Conference on File and Storage Technologies*, San Jose, CA, 2011, pp. 77-90.

[10] W. K. Ng, Y. Wen, H. Zhu, Private Data Deduplication Protocols in Cloud Storage, *27th Annual ACM Symposium on*

Applied Computing, Riva, Italy, 2012, pp. 441-446.

[11] Y. Gu, D. Wang, C. Liu, DR-Cloud: Multi-cloud Based Disaster Recovery Service, *Tsinghua Science and Technology*, Vol. 19, No. 1, pp. 13-23, February, 2014.

[12] J. Li, Y. K. Li, X. Chen, P. P. Lee, W. Lou, A Hybrid Cloud Approach for Secure Authorized Deduplication, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 26, No. 5, pp. 1206-1216, May, 2015.

[13] Si Yu, Xiaolin Gui, Xuejun Zhang, Jiancai Lin, Min Dai, Detecting Cache-Based Side Channel Attacks in the Cloud: An Approach with Cascade Detection Mode, *Journal of Internet Technology*, Vol. 15, No. 6, pp. 903-915, November, 2014.

[14] D. Harnik, B. Pinkas, A. Shulman-Peleg, Side Channels in Cloud Services: Deduplication in Cloud Storage, *IEEE Security and Privacy*, Vol. 8, No.6, pp. 40-47, November, 2010.

[15] C. Yang, J. Ma, J. Ren, Provable Ownership of Encrypted Files in De-duplication Cloud Storage, *Ad Hoc & Sensor Wireless Networks*, Vol. 26, No. 1-4, pp. 43-72, January, 2015.

[16] J. Stanek, A. Sorniotti, E. Androulaki, L. Kencl, A Secure Data Deduplication Scheme for Cloud Storage, *International Conference on Financial Cryptography and Data Security*, Christ church, Barbados, 2014, pp. 99-118.

[17] Z. Yan, M. Wang, Y. Li, A. V. Vasilakos, Encrypted Data Management with Deduplication in Cloud Computing, *IEEE Cloud Computing*, Vol. 3, No. 2, pp. 28-35, March, 2016.

[18] C. Yang, J. Ren, J. Ma, Provable Ownership of Files in Deduplication Cloud Storage, *Security and Communication Networks*, Vol. 8, No. 14, pp. 2457-2468, September, 2015.

[19] J. Li, X. Chen, M. Li, J. Li, P. P. C. Lee, W. Lou, Secure Deduplication with Efficient and Reliable Convergent Key Management, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 25, No. 6, pp. 1615-1625, June, 2014.

[20] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, M. Theimer,P. Simon, Reclaiming Space from Duplicate Files in a Serverless Distributed File System, *22nd International Conference on Distributed Computing Systems*, Vienna, Austria, 2002, pp. 617-624.

[21] Z. Yan, W. Ding, X. Yu, H. Zhu, R. H. Deng, Deduplication on Encrypted Big Data in Cloud, *IEEE Transactions on Big Data*, Vol. 2, No. 2, pp. 138-150, June, 2016.

[22]M. Bellare, S. Keelveedhi, T. Ristenpart, Message - Locked Encryption and Secure Deduplication, *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Athens, Greece, 2013, pp. 296-312.

[23] M. Bellare, S. Keelveedhi, Interactive Message - Locked Encryption and Secure Deduplication, *International Conference on Practice and Theory in Public-Key Cryptography*, Gaithersburg, Maryland, 2015, pp. 516-538.

[24] M. Bellare, S. Keelveedhi, T. Ristenpart, DupLESS: Server-Aided Encryption for Deduplicated Storage, *IACR Cryptology ePrint Archive*, Report No.2013/429, August, 2013.

[25] C. M. Yu, C. Y. Chen, H. C. Chao, Proof of Ownership in Deduplicated Cloud Storage with Mobile Device Efficiency, *IEEE Network*, Vol. 29, No. 2, pp. 51-55, March, 2015.

[26] S. Halevi, D. Harnik, B. Pinkas, A. Shulman - Peleg, Proofs of Ownership in Remote Storage Systems, *ACM Conference on Computer and Communications Security*, Chicago, IL, 2011, pp. 491-500.

[27] N. Kaaniche, M. Laurent, A Secure Client Side Deduplication Scheme in Cloud Storage Environments, *6th International Conference on New Technologies, Mobility and Security*, Dubai, UAE, 2014, pp. 1-7.

[28] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved proxy Re-encryption Schemes with Applications to Secure distributed storage, *ACM Transactions on Information and System Security*, Vol.9, No. 1, pp. 1-30, February, 2006.

[29] A. Juels, B. S. Kaliski Jr, PORs: Proofs of Retrievability for Large Files, *14th ACM conference on Computer and communications security*, Alexandria, VA, 2007, pp. 584-597.

[30] Q. Zheng, S. Xu, Secure and Efficient Proof of Storage with Deduplication, *Second ACM Conference on Data and Application Security and Privacy*, San Antanio, TX, 2012, pp. 1-12.

[31] F. Armknecht, C. Boyd, G. T. Davies, K. Gjosteen, M. Toorani, Side Channels in Deduplication: Trade-offs between Leakage and Efficiency, *Asia Conference on Computer and Communications Security*, Abu Dhabi, UAE, 2017, pp. 266-274.

[32] O. Heen, C. Neumann, L. Montalvo, S. Defrance, Improving the Resistance to Side-channel Attacks on Cloud Storage Services, *5th International Conference on New Technologies, Mobility and Security*, Istanbul, Turkey, 2012, pp. 1-5.

[33] Y. Shin, K. Kim, Differentially Private Client- Side Data Deduplication Protocol for Cloud Storage Services, *Security and Communication Networks*, Vol. 8, No. 12, pp.2114-2123, August, 2015.

[34] S. Lee, D. Choi, Privacy-preserving Cross-user Source-based Data Deduplication in Cloud Storage, *International Conference on ICT Convergence*, Jeju, Korea, 2012, pp. 329-330.

[35] B. Wang, W. Lou, Y. T. Hou, Modeling the Side-channel Attacks in Data Deduplication with Game Theory, *IEEE Conference on Communications and Network Security*, Florence, Italy, 2015, pp. 200-208.

[36] P. Zahra, C. Kang-Cheng, Y. Chia-Mu, C. Mauro, RARE: Defeating Side Channels Based on Data-Deduplication in Cloud Storage, *International Conference on Computer Communications: Cloud Computing Systems, Networks, and Applications*, Honolulu, HI, 2018, pp. 15-19.

[37] G. H. Yu, Research on Mobile Internet Big Data Detecting Method for the Redundant Data, *International Journal of Internet Protocol Technology*, Vol. 11, No.1, pp.29-37, April 2018.

Biographies

S. Annie Joice is a part-time Ph.D. student and working as Assistant Professor in the Department of Computer Science and Engineering in Government College of Engineering, Srirangam. She received B.E Computer Science and Engineering from Bharathidasan University, Tiruchirappalli and received M.E. Computer Science and Engineering from Anna University, Chennai. Her research interests include Security, Cloud Computing,and Automata Theory.

M. A. Maluk Mohamed obtained his Ph.D. degree from IIT Madras, Chennai in the year 2006. He is a Professor of M.A.M. College of Engineering, Affiliated to Anna University, Chennai. He has(co-)authored over 80 research articles published in refereed journals and conferences, and is a frequent invited speaker at conferences and institutions all over India. His current research focus is on Distributed Computing, Mobile Computing, wireless Sensor Networks, Cluster Computing, Grid Computing, etc. He is a member of the ACM, IEEE, ISA, IARCS and life member of the Computer Society of India.