ISSN:0975-9646

Protected Authorized Deduplication on Cloud Using Hybrid Cloud Approach

Dr.V.Goutham[1], G.Shivakrishna[2], M.Prathyusha[3]

[1,2,3]*Department of Computer and Engineering, Teegala Krishna Reddy Engineering College, Meerpet, Telangana, India*

Abstract- **Data deduplication involves finding and removing duplication within data without compromising its integrity. In the view of protecting the confidentiality of sensitive data, a convergent encryption technique has been brought into existence to encrypt the data before outsourcing. Comparatively with the conventional deduplication systems, the differential privileges of users are further considered induplicate check besides the data itself. Based on hybrid cloud architecture, many deduplication constructions supporting authorized duplicate check. In the proposed authorized duplicate check scheme a new prototype has been implemented and provided with experimental evaluation reports. In this paper we discussed and addressed the problem of privacy preserving deduplication in cloud computing and supportive factors like differential authorization, authorized duplicate check, uncountability of the file token.**

Keywords- **Fidelity, Confidentiality, Authorized duplicate check scheme, Authorized data deduplication, Convergent encryption.**

I. INTRODUCTION

Cloud computing can help enterprises progress the creation and delivery of IT solutions by providing them with access to services in a cost-effective and flexible manner. Clouds can be classified into three categories, based on their convenience precincts and the deployment model. They are: Public Cloud, Private Cloud and Hybrid Cloud[1].

A public Cloud is made available in a pay-as-you-go manner to the general public users irrespective of their original association. A private Cloud's usage is restricted to members, employees, and trusted partners of the organization. A hybrid Cloud enables the use of private and public Cloud in a seamless manner. Cloud computing applications span many domains, including business, technology, government, health care, smart grids, intelligent transportation networks, life sciences, disaster management, automation, data analytics, and consumer and social networks. Various models for the creation, deployment, and release of these applications as Cloud services have emerged. To compose data management scalable in cloud computing, deduplication has been a well-known technique and has fascinated more and more attention in recent times[1]. Data deduplication is a dedicated data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to condense the number of bytes that must be sent. As an alternative of observance multiple data copies with the same content, deduplication pure redundant data by keeping only one physical copy and referring other redundant data to that copy. Deduplication can take place at either the file level or the block level. For file-level deduplication, it eliminates duplicate copies of the same file. It eliminates duplicate blocks of data that occur in non-identical files which takes place at the block level[3].

Data deduplication brings a lot of benefits, security and privacy concerns arise as users' sensitive data are vulnerable to both insider and outsider attacks. Convergent encryption allows the cloud to perform deduplication on the cipher texts and the proof of ownership thwart the unauthorized user to access the file. Earlier deduplication systems cannot support differential authorization duplicate check, which is important in many applications. In such an authorized deduplication system, each user is issued a set of privileges during system initialization. Each file uploaded to the cloud is also circumscribed by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files[6]. Before submitting his duplicate check request for a file, the user needs to take this file and own privileges as inputs. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched privilege stored in cloud. Let us consider an example, in a company, many different privileges will be assign to employees and in order to save cost and efficiently management, the data will be stimulated to the storage server provider (S-CSP) in the public cloud with specified privileges and the deduplication technique will be applied to store only one copy of the same file. Because of privacy consideration, some files will be encrypted and allowed the duplicate check by employees with precise privileges to comprehend the access control.

II. RELATED WORK

Conventional encryption, is incompatible with data deduplication while providing data confidentiality. This type of encryption requires different users to encrypt their data with their own keys[2]. Hence, identical data copies of different users will lead to different ciphertexts, making deduplication impossible. Convergent encryption has been proposed to enforce data confidentiality while making deduplication feasible[5]. It encrypts/decrypts a data copy with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the cipher text to the cloud. In view of the fact that the encryption operation is deterministic and is derived from the data content, identical data copies will generate the same convergent key and hence the same ciphertext. To put a stop to unauthorized access, a secure proof of ownership (POW) protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the equivalent data owners with their convergent keys[7].

Secure deduplication: With the advent of cloud computing, secure data deduplication has attracted much attention recently from research community. Yuan and Yu proposed a deduplication system in the cloud storage to reduce the storage size of the tags for integrity check. To increase the security of deduplication and protect the data confidentiality, Bellare et al. showed how to defend message into unpredictable message[4]. In their system, another third party called key server is introduced to generate the file tag for duplicate check. Stanek et al. presented a novel encryption scheme that provides differential security for popular data and unpopular data. For popular data that are not particularly sensitive, the conventional encryption is performed[1]. Another two layered encryption scheme with stronger security while supporting deduplication is proposed for detested data. In this way, they achieved better tradeoff between the efficiency and security of the outsourced data. Li et al. addressed the key-management issue in block-level deduplication by distributing these keys across multiple servers after encrypting the files[9].

Convergent encryption: Convergent encryption ensures data privacy in deduplication. Bellare et al. formalized this primitive as message-locked encryption, and explored its application in space-efficient secure outsourced storage. Xu et al. also addressed the problem and showed a secure convergent encryption for efficient encryption, without considering issues of the key-management and block level deduplication[8]. There are also several implementations of convergent implementations of different convergent encryption variants for secure deduplication It is known that some commercial cloud storage providers, such as Bitcasa, also deploy convergent encryption. Proof of ownership Halevi et al. proposed the notion of “proofs of ownership” for deduplication systems, such that a client can efficiently prove to the cloud storage server that he/she owns a file without uploading the file itself. Several PoW constructions based on the Merkle-Hash Tree are proposed to enable client-side deduplication, which include the bounded leakage setting. Pietro and Sorniotti proposed another efficient PoW scheme by choosing the projection of a file onto randomly selected bit positions as the file proof[8]. Recently, Ng et al. extended PoW for encrypted files, but they do not address how to minimize the key management overhead. Recently, Bugiel et al. provided an architecture consisting of twin clouds for secure outsourcing of data and arbitrary computations to an untrusted commodity cloud. Zhang et al[6]. also presented the hybrid cloud techniques to support privacy-aware data intensive computing[2]. Here, to address the authorized deduplication problem over data in public cloud, the security model of these systems is similar to those where the private cloud is assumed to be honest but curious.

III. SYSTEM DESIGN

In the present system, three entities are defined. They are users, private cloud and S-CSP in public cloud. The S-CSP performs deduplication by checking if the contents of two files are the same and stores only one of them. The access right to a file is defined based on a set of privileges[10]. For example, we may define a role-based privilege according to job positions or we may define a time-based privilege that specifies a valid time period within which a file can be accessed. A user, say Alice, may be assigned two privileges “Director” and “access right valid on 2014-01-01”, so that she can access any file whose access role is “Director” and accessible time period starts from 2014-01-01. Each privilege is represented in the form of a short message called token. Each file is associated with some file tokens, which denote the tag with specified privileges. A user computes and sends duplicate check tokens to the public cloud for authorized duplicate check[11]. Users have access to the private cloud server, a semi trusted third party which will aid in performing deduplicable encryption by generating file tokens for the requesting users. Users are also provisioned with per user encryption keys and credentials. Here it is considered the file-level deduplication for simplicity. We also refer a data copy to be a whole file and file-level deduplication which eliminate the storage of any redundant files. Specifically, to upload a file, a user first performs the file-level duplicate check. If the file is a duplicate, then all its blocks must be duplicates as well or else, the user further performs the block-level duplicate check and identifies the unique blocks to be uploaded. Each data copy is associated with a token for the duplicate check.

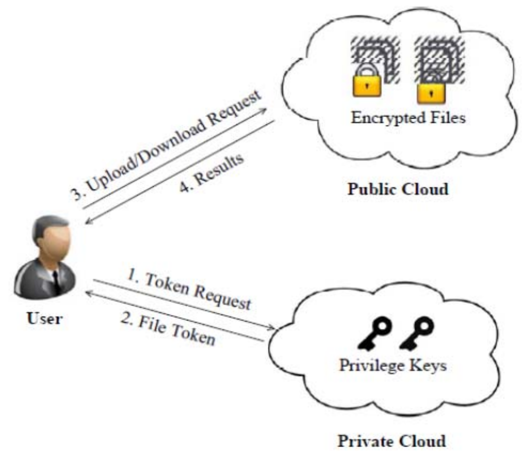

Fig.1: Architecture for authorized deduplication.

S-CSP. This is an entity that provides a data storage service in public cloud. The S-CSP provides the data outsourcing service and stores data on behalf of the users. The S-CSP eliminates the storage of redundant data via deduplication and keeps only unique data to reduce the storage cost. Assuming that S-CSP is always online and has abundant storage capacity and computation power.

Data users: A user is an entity that requests to outsource data storage to the S-CSP and access the data later. In a storage system sustaining deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users. Each user is issued a set of privileges in the setup of the system in the authorized deduplication system. Each file is protected with the convergent encryption key and privilege keys to realize the authorized deduplication with differential privileges.

Private cloud: A new entity introduced for facilitating user's secure usage of cloud service when compared with the traditional deduplication architecture in cloud computing. Specifically, since the computing resources at data user/owner side are restricted and the public cloud is not fully trusted in practice, private cloud is able to provide data user/ owner with an execution environment and infrastructure working as an interface between user and the public cloud. The private keys for the privileges are managed by the private cloud, who answers the file token requests from the users. The interface offered by the private cloud allows user to submit files and queries to be securely stored and computed correspondingly [12].

IV. Proposed construction

Resourcefully solving the problem of deduplication with discrepancy privileges in cloud computing, it is considered a hybrid cloud architecture consisting of a public cloud and a private cloud. Contrasting with existing data deduplication systems, the private cloud is involved as a proxy to allow data owner/users to securely perform duplicate check with differential privileges. Such an architecture is practical and has attracted much attention from researchers. The data owners only outsource their data storage by utilizing public cloud while the data operation is managed in private cloud[10]. A new deduplication system supporting differential duplicate check is proposed under this hybrid cloud architecture where the S-CSP resides in the public cloud. The user is only allowed to perform the duplicate check for files marked with the corresponding privileges. Additionally, the present system is enhanced in security. Specifically, an advanced scheme to support stronger security by encrypting the file with differential privilege keys is introduced. In this way, the users without corresponding privileges cannot perform the duplicate check. Furthermore, such unauthorized users cannot decrypt the ciphertext even get together with the S-CSP.

Security analysis exhibit that the system is secure in terms of the definitions specified in the proposed security model [11]. A prototype of the proposed authorized duplicate check is implemented and conduct test bed experiments to evaluate the overhead of the prototype. The overhead is minimal compared to the normal convergent encryption and file upload operations. Addressing the problem of privacy-preserving deduplication in cloud computing it is proposed a new deduplication system supporting for following factors:

Differential authorization: Each authorized user is able to get his/her individual token of his file to perform duplicate check based on his privileges. Under this supposition, any user cannot generate a token for duplicate check out of his privileges or without the aid from the private cloud server.

Authorized duplicate check: Authorized user is able to use his/her individual private keys to generate query for certain file and the privileges he/she owned with the help of private cloud, while the public cloud performs duplicate check directly and tells the user if there is any duplicate[11]. The security requirements lie in two folds, including the security of file token and security of data files. For the security of file token, two aspects are defined as unforgeability and indistinguishability of file token.

Uncountability of file token/duplicate-check token: Unauthorized users without appropriate privileges or file should be prevented from getting or generating the file tokens for duplicate check of any file stored at the S-CSP. The users are not allowed to collude with the public cloud server to break the uncountability of file tokens. In this system, the S-CSP is honest but curious and will honestly perform the duplicate check upon receiving the duplicate request from users. The duplicate check token of users should be issued from the private cloud server in the present scheme.

Indistinguishability of file token/duplicate-check token: It requires that any user without querying the private cloud server for some file token, he cannot get any useful information from the token, which includes the file information or the privilege information.

Data confidentiality: Unauthorized users without appropriate privileges or files, including the S-CSP and the private cloud server, should be prevented from access to the underlying plaintext stored at S-CSP. The goal of the adversary is to retrieve and recover the files that do not belong to them.

4.1 **Secure Deduplication Systems**

To support authorized deduplication, the tag of a file F will be determined by the file F and the privilege. To support authorized access, a secret key kp will be bounded with a privilege p to generate a file token. If a file has been uploaded by a user with a duplicate token, then a duplicate check sent from another user will be successful if and only if he also has the file F and privilege p.

4.1.1. File uploading: Suppose that a data owner with privilege set wants to upload and share a file F with users who have the privilege set. The user computes and sends S-CSP the file token.

4.1.2. File retrieving: Suppose a user wants to download a file F. It first sends a request and the file name to the S-CSP. Upon receiving the request and file name, the S-CSP will check whether the user is eligible to download F. If failed, the S-CSP sends back a signal to the user to indicate the download failure. Otherwise, the S-CSP returns the corresponding cipher text CF. Upon receiving the encrypted data from the S-CSP, the user uses the key k_F stored locally to recover the original file F.

To solve the problems of the construction another advanced deduplication system supporting authorized duplicate check is proposed. In this new deduplication system, hybrid cloud architecture is introduced to solve the problem[9]. The private keys for privileges will not be issued to users directly, which will be kept and managed by the private cloud server instead. In this way, the users cannot share these private keys of privileges in this proposed construction, which means that it can prevent the privilege key sharing among users in the above straightforward construction.

To get a file token, the user needs to send a request to the private cloud server. The intuition of this construction can be described as follows. To perform the duplicate check for some file, the user needs to get the file token from the private cloud server. The private cloud server will also check the user's identity before issuing the corresponding file token to the user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file[10]. Based on the results of duplicate check, the user uploads this file. Before giving the construction of the deduplication system, given two privileges. This kind of a generic binary relation definition could be instantiated based on the background of applications, such as the common hierarchical relation. More precisely, in a hierarchical relation, for example, in an enterprise management system, three hierarchical privilege levels are defined as Director, Project lead, and Engineer, where Director is at the top level and Engineer is at the bottom level.

The proposed deduplication system is as follows:

System setup: The privilege universe P is defined as a symmetric key k_{pi} for each p_i will be selected and the set of keys will be sent to the private cloud. An identification protocol is also defined, where Proof and Verify are the proof and verification algorithm respectively. Furthermore, each user U is assumed to have a secret key to perform the identification with servers[11]. Assume that user U has the privilege set PU. It also initializes a protocol for the file ownership proof. The private cloud server will maintain a table which stores each user's public information and its corresponding privilege set PU.

We design and implement a new system which could protect the security for predicatable message. The main idea of this technique is that the novel encryption key generation algorithm. For simplicity, we will use the hash functions to define the tag generation functions and convergent keys in this section. In traditional convergent encryption, to support duplicate check, the key is derived from the file F by using some cryptographic hash function. To avoid the deterministic key generation, the encryption key k_F for file F in the system will be generated with the aid of the private key cloud server with privilege key kp. The file F is encrypted with another key k, while k will be encrypted with k_F. In this way, both the private cloud server and S-CSP cannot decrypt the cipher text. Furthermore, it is semantically secure to the S-CSP based on the security of symmetric encryption. For S-CSP, if the file is unpredicatable, then it is semantically secure too[12]. The details of the scheme, which has been instantiated with hash functions for simplicity, are described below.

System setup: The privilege universe P and the symmetric key kp_i for each will be selected for the private cloud as above. An identification protocol is also defined. The proof of ownership is instantiated by hash functions. The private cloud server maintains a table which stores each user's identity and its corresponding privilege.

File uploading: Suppose that a data owner with privilege p wants to upload and share a file F with users whose privilege belongs to the set. The data owner performs the identification and sends to the private cloud server. If a file duplicate is found, the user needs to run the protocol with the SCSP to prove the file ownership. If the proof is also passed, the user will be provided a pointer for the file. Otherwise, if no duplicate is found, a proof from the S-CSP will be returned. The user sends the privilege set as well as the proof to the private cloud server. Upon receiving the request, the private cloud server verifies the signature. If it is passed, the private cloud server will compute, which will be returned to the user and S-CSP, respectively. Then, the user computes the encryption, where k is random key. The key k will be encrypted into ciphertext using a symmetric encryption algorithm.

IV. IMPLEMENTATION:

Token generation and deduplication along the file upload process

A Client program is used to model the data users to carry out the file upload process. A Private Server program is used to model the private cloud which manages the private keys and handles the file token computation. A Storage Server program is used to model the S-CSP which stores and deduplicates files. We implement cryptographic operations of hashing and encryption with the OpenSSL library. We also implement the communication between the entities based on HTTP, users can issue HTTP Post requests to the servers. Our implementation of the Client provides the following function calls to support token generation and deduplication along the file upload process.

FileTag(File): It computes SHA-1 hash of the File as File Tag.

TokenReq(Tag, UserID): It requests the Private Server for File Token generation with the File Tag and User ID.

DupCheckReq(Token): It requests the Storage Server for Duplicate Check of the File by sending the file token received from private server.

ShareTokenReq(Tag, {Priv.}): It requests the Private Server to generate the Share File Token with the File Tag and Target Sharing Privilege Set.

FileEncrypt(File): It encrypts the File with Convergent Encryption using 256-bit AES algorithm
in cipher block chaining (CBC) mode, where the convergent key is from SHA-256 Hashing of the file.

FileUploadReq(FileID, File, Token): It uploads the File Data to the Storage Server if the file is Unique and updates the File Token stored.

Our implementation of the Private Server includes corresponding request handlers for the token generation and maintains a key storage with Hash Map.

TokenGen(Tag, UserID): It loads the associated privilege keys of the user and generate the token
with HMAC-SHA-1 algorithm.

ShareTokenGen(Tag, {Priv.}): It generates the share token with the corresponding privilege keys of the sharing privilege set with HMAC-SHA-1 algorithm[12].

Implementation of the Storage Server provides deduplication and data storage with following handlers and maintains a map between existing files and associated token with Hash Map.
DupCheck(Token): It searches the File to Token Map for Duplicate.
FileStore(FileID, File, Token): It stores the File on Disk and updates the Mapping.

VI. EVALUATION: TEST ANALYSIS AND REPORTS

Focusing on comparing the overhead induced by authorization steps, includes file token generation and share token generation, against the convergent encryption and file upload steps[12]. We evaluate the overhead by varying different factors, including 1) File Size, 2) Number of Stored Files, 3) Deduplication Ratio, 4) Privilege Set Size.

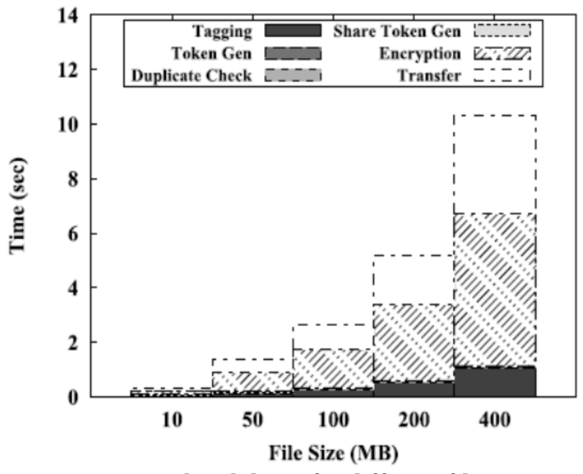

Fig.2: Time breakdown for different file size.

Evaluation of the prototype with a real-world workload based on VM images is done. The experiments with three machines equipped with an Intel Core-2-Quad 2.66 GHz Quad Core CPU, 4 GB RAM and installed with Ubuntu 12.04 32-Bit Operation System[11]. The machines are connected with 1 Gbps Ethernet network. We break down the upload process into six steps, 1) Tagging, 2) Token Generation, 3) Duplicate Check, 4) Share Token Generation, 5) Encryption, 6)Transfer. For each step, we record the start and end time of it and therefore obtain the breakdown of the total time spent.

File Size:
To evaluate the effect of file size to the time spent on different steps, we upload 100 unique files of particular file size and record the time break down. Using the unique files enables us to evaluate the worst-case scenario where we have to upload all file data. The average time of the steps from test sets of different file size are plotted. The time spent on tagging, encryption, upload increases linearly with the file size, since these operations involve the actual file data and incur file I/O with the whole file[13].

Number of Stored Files:
To evaluate the effect of number of stored files in the system, we upload 10,000 10 MB unique files to the system and record the breakdown for every file upload[11]. Every step remains constant along the time. Token checking is done with a hash table and a linear search would be carried out in case of collision. Despite of the possibility of a linear search, the time taken in duplicate check remains stable due to the low collision probability.

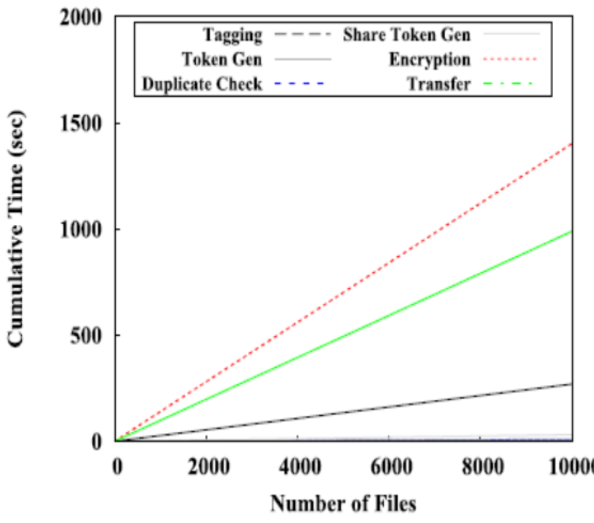

Fig.3: Time breakdown for different number of stored files

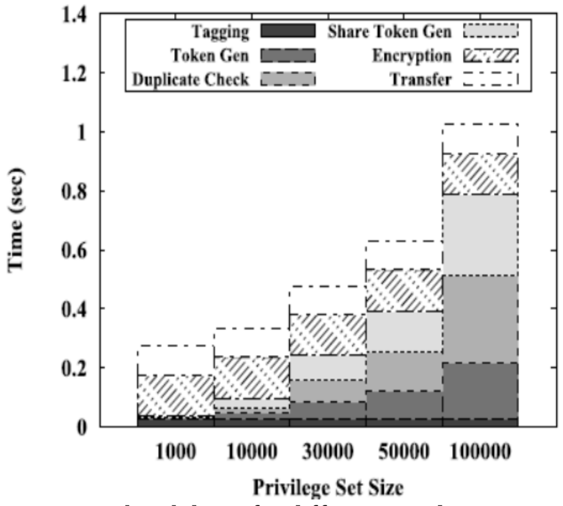

Fig. 5: Time breakdown for different privilege set size.
Deduplication Ratio:

To evaluate the effect of the deduplication ratio, two unique data sets are prepared, each of which consists of 50 100 MB files[13]. We first upload the first set as an initial upload. For the second upload, we pick a portion of 50 files, according to the given deduplication ratio, from the initial set as duplicate files and remaining files from the second set as unique files. Total time spent on uploading the file with deduplication ratio at 100 percent is only 33.5 percent with unique files.

Privilege Set Size:
To evaluate the effect of privilege set size, we upload 100 10 MB unique files with different size of the data owner and target share privilege set size.

VII. Conclusion

The conception of authorized data deduplication was projected to protect the data security by including differential privileges of users in the duplicate check. Several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys, has been presented. Security analysis exhibit that the system is secure in terms of the definitions specified in the proposed security model. Schemes presented in this architecture are secure in terms of insider and outsider attacks. A prototype of proposed authorized duplicate check scheme experiments and their reports presented on this prototype. The authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

References

[1] OpenSSL Project, (1998). [Online]. Available: http://www. openssl.org/

[2] P. Anderson and L. Zhang, "Fast and secure laptop backups With encrypted de-duplication," in Proc. 24th Int. Conf. Large Installation Syst. Admin., 2010, pp. 29–40.

[3] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: Serveraided encryption for deduplicated storage," in Proc. 22nd USENIX Conf. Sec. Symp., 2013, pp. 179–194.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in Proc. 32nd Annu. Int. Conf. Theory Appl. Cryptographic Techn., 2013, pp. 296–312.

[5] M. Bellare, C. Namprempre, and G. Neven, "Security proofs For identity-based identification and signature schemes," J. Cryptol., vol. 22, no. 1, pp. 1–61, 2009.

[6] M. Bellare and A. Palacio, "Gq and schnorr identification Schemes: Proofs of security against impersonation under active and concurrent attacks," in Proc. 22nd Annu. Int. Cryptol. Conf. Adv. Cryptol., 2002, pp. 162–177.

[7] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider, "Twin clouds: An architecture for secure cloud computing," in Proc. Workshop Cryptography Security Clouds, 2011, pp. 32–44.

[8] C. Ng and P. Lee, "Revdedup: A reverse deduplication storage system optimized for reads to latest backups," in Proc. 4th Asia-Pacific Workshop Syst., http://doi.acm.org/10.1145/2500727. 2500731, Apr. 2013.

[9] W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage," in Proc. 27th Annu. ACM Symp. Appl. Comput., 2012, pp. 441–446.

[10] R. D. Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," in Proc. ACM Symp. Inf., Comput. Commun. Security, 2012, pp. 81–82.

[11] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui, "A secure cloud backup system with assured deletion and version control," in Proc. 3rd Int. Workshop Secutiry Cloud Comput., 2011, pp. 160–167.

[12] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," IEEE Comput., vol. 29, no. 2, pp. 38–47, Feb. 1996.

[13] J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," IACR Cryptology ePrint Archive, 2013:149, 2013.

Authors:

[1] Dr V. Goutham is a Professor and Head of the Department of computer Science and Engineering at TKR Engineering College affiliated to J.N.T.U Hyderabad. He received M.Tech from Andhra University and B.Tech from J.N.T.U Hyderabad. He worked for various MNC Companies in Software Testing and Quality as Senior Test Engineer. His research interests are Software Reliability Engineering, software testing, software Metrics, and cloud computing.

[2] G.Shivakrishna is a Assistant Professor in the Department of computer Science and Engineering at TKR Engineering College affiliated to J.N.T.U Hyderabad.

[3] Ms. M.Prathyusha Department of computer Science and Engineering at TKR Engineering College affiliated to J.N.T.U Hyderabad..