International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 02 Issue: 03 | June-2015 www.irjet.net p-ISSN: 2395-0072

IMPLEMENTATION OF HYBRID CLOUD APPROACH FOR SECURE AUTHORIZED DEDUPLICATION

JADAPALLI NANDINI[1], RAMIREDDY NAVATEJA REDDY[2]

[1] *M.Tech student, Narayana Engineering College, Nellore, A.P, India*
[2] *Assistant professor, Dept of CSE, Narayana Engineering College, Nellore, A.P, India*

Abstract- *This paper represents that, many techniques are using for the elimination of duplicate copies of repeating data, from that techniques, one of the important data compression technique is data duplication. Many advantages with this data duplication, mainly it will reduce the amount of storage space and save the bandwidth when using in cloud storage. To protect confidentiality of the sensitive data while supporting de-duplication data is encrypted by the proposed convergent encryption technique before out sourcing. Problems authorized data duplication formally addressed by the first attempt of this paper for better protection of data security. This is different from the traditional duplication systems. The differential privileges of users are further considered in duplicate check besides the data itself. In hybrid cloud architecture authorized duplicate check supported by several new duplication constructions. Based on the definitions specified in the proposed security model, our scheme is secure. Proof of the concept implemented in this paper by conducting test-bed experiments.*

Key Words: de-duplication, hybrid cloud, authorized duplicate check, confidentiality, encryption.

1. INTRODUCTION

Unlimited "virtualized" resources to users as services across the whole internet providing by the cloud computing to hide platforms and implementation details. Highly available storage and massively parallel computing resources providing by the cloud services at low costs. Cloud computing widely spread in the world, maximum amount of data stored in the clouds and shred by the users with specified rights, which define as access rights of the stored data. One of the critical challenge of cloud storage services is the management of the duplication is one of the best technique to make the data management in the cloud computing. It has attracted more and more attention recently. In the data storage to reduce the data copies we go for duplication techniques. This duplication technique is a data compression technique[2]. The technique is used improve storage utilization and can also applied for network data transfer to reduce the number of byte that must be sent. De-duplication eliminates redundant data to reduce multiple data copies with the same content. Duplication only keeps one physical copy and referring other redundant data to that copy. Either the file level or block level, de-duplication can take place. Same file duplicate copies eliminated in file level de-duplication. In non-identical files, blocks of data that occur, this blocks of data eliminate with the block de-duplication. The detailed system architecture is shown in figure 1.1.

Although data de-duplication brings a lot of advantages, security and privacy concerns arise as users sensitive data are susceptible to both the insider and outsider attacks. When compares the traditional encryption with data duplication. It will provide data confidentiality. In the traditional encryption requires different users to encrypt data with their own keys. Thus identical copies of different users will lead to different cipher texts, making de-duplication impossible. One of the new technique has been proposed to encrypt data confidentiality while making de-duplication feasible, i,e convergent encryption. This convergent encrypt provides one convergent key to encrypt/decrypt the data, which is obtained by computing the cryptographic hash value of the content of the data copy. After completion of key generation and data encryption, users retain the keys and send the cipher text to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies generate the same convergent key and hence the same cipher text. A secure proof of ownership protocol [11] is also required to provide the proof that the user indeed owns. This is all for prevent unauthorized access, the same file duplicate will found, this process will occur. A pointer from the server will provide to user, after the proof submission, who are having the subsequent file without needing upload the same file. The encrypted file can be downloaded by the user and also decrypted by the corresponding data users with their convergent keys. Thus, convergent encryption allows the cloud to perform de-duplication on the cipher texts and the proof of ownership prevents the unauthorized user to access the file.

"Differential authorized de-duplication check" cannot supported by the previous de-duplication systems. With the authorized de-duplication system, each user issued a set of the privileges during system initialization. To specify which type of user is allowed to perform the duplication check and access the files is decided by the uploading each file to the cloud and is also bounded by the set of privileges. The user have to take the file and the own privileges as inputs, to submit before of the user duplication check request for the same file. If only, copy of the file and matched privilege stored in cloud, then only the user gets the duplicate of the same file.

1.1 Contributions:

The main problems in the cloud computing is de-duplication with differential privileges. The main aim of this paper is to solve this problem. For this we go with different type of architecture, which is having public cloud and private cloud i.e., "Hybrid Cloud Architecture". Private cloud is main part, that is involved as the substitution to allow data owner/users to securely perform de-duplication check with differentials privileges. The data owners/users only outsource their data storage by using public cloud and data operation is managing in private cloud. Differential duplication check is proposed under the hybrid cloud architecture separated by a new de-duplication system. A user only with corresponding privilege only marked files has been allowed to perform de-duplication.

We enhance our system in security for future scope. Specifically, we present an advanced scheme to support stronger security by encrypting the file with differential privilege keys. Without the privilege key the duplication check cannot perform. Such type of unauthorized users can decrypt the data even conspire with the S-CSP security analysis demonstrates that our system is secure in terms of the definitions specified in this model.

Table 1: Notations appeared in this paper

Acronym	Description
S-CSP	Storage-cloud service provider
PoW	Proof of Ownership
$(pk_{U,}sk_{U,})$	User's public and secret key pair
$k_{F,}$	Convergent encryption key for file F
$P_{U,}$	Privilege set of a user U
P_F	Specified privilege set of a file F
$\emptyset'_{F,p}$	Token of the file F with privilege p

2. PRELIMINARIES:

In this we go through with the notations used in this paper. Analyze the secure primitives used in our secure duplication.

- Symmetric Encryption:
  It uses a common secret key k to encrypt and decrypt information. A symmetric encryption scheme consists of three primitive functions:

a. KeyGenSE(1^{λ}) $\rightarrow\kappa$ is the key generation algorithm that generates κ using security parameter 1^{λ};

b. EncSE(κ,M) $\rightarrow C$ is the symmetric encryption algorithm that takes the secret κ and message M and then outputs the cipher text C; and

c. DecSE(κ,C) $\rightarrow M$ is the symmetric decryption algorithm that takes the secret κ and cipher text C and then outputs the original message M.

Convergent Encryption:

With this convergent encryption [4], [8] we get secure confidentiality of de-duplication. Data owner gets convergent key from each original data copy and encrypts data copy with the convergent key. A tag is also provide to the user with the data copy, tag will be used to detect duplicates. If two data copies are same, then their tags are same. To identify and check the duplicates, the user first sends a tag to the server side to check. Server will replies, if the identical copy has been already stored or not. Both (confidentiality check and tag) are independently derived. Tag cannot used to reduce the convergent key and compromise data confidentiality. Tag and it's encrypted data copy will be stored in the server side. With the four

primitive functions we can define the convergent encryption scheme.

- KeyGenCE(M) → K is the key generation algorithm that maps a data copy M to a convergent key K;
- EncCE(K,M) → C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M as inputs and then outputs cipher text C;
- DecCE(K,C) → M is the decryption algorithm that takes both the cipher text C and the convergent key K as inputs and then outputs the original data copy M; and
- TagGen(M) → $T(M)$ is the tag generation algorithm that maps the original data copy M and outputs a tag $T(M)$.

Proof of ownership:

Enable the users to provide their ownership of data copies to the storage server we choose proof of ownership. Proof of ownership is implemented as an interactive algorithm run by a power and verifier. From a data copy of M, the verifier derives a short value Φ(M). To prove the ownership of the data copy M, the user needs to send Φ to the verifier such that $\Phi' = \Phi(M)$. The formal security definition for PoW roughly follows the threat model in a content distribution network, where an attacker does not know the entire file, but has accomplices who have the file. The accomplices follow the "bounded retrieval model", such that they can help the attacker obtain the file, subject to the constraint that they must send fewer bits than the initial min-entropy of the file to the attacker [11].

Identification protocol:

With two phases we can describe the identification protocol, Proof and Verify. In the stage of proof, a user U demonstrates his identity to a verifier by performing the some identification proof related to his identity. Private sk_U is the input of the prover/user i.e sensitive information such as private key of a public key in his certificate, credit card number, etc. These types of numbers cannot share with others. With the help of input of public information pk_U related to sk_U, the verifier perform the verification. At the end of protocol, the verifier output either accept or not to denote whether the proof is passed or not. Different types of identification protocols are there like, certificate based and identification based identification [5], [6].

3. SYSTEM MODEL

3.1 Secure Duplication with Hybrid Architecture:

By using the duplication technique, to store the data who will use S-CSP are consisted as group of affiliated client at high level. The main aim is enterprise all the network. To set the data back up and disaster recovery applications for reduce the storage space. We frequently go for de-duplication. Such systems are widespread and are often more suitable to user file backup and synchronization applications than richer storage abstractions.

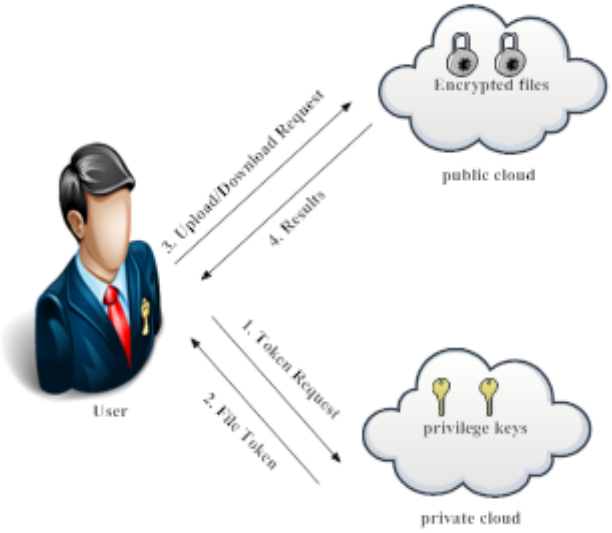

Fig-1: Working of authorized de-duplication

There are three entities define in our system as shown in figure 1, those are,

- Users
- Private cloud
- S-CSP in public cloud

De-duplication performed by S-CSP by checking if the contents of two files are the same and stores only one of them.

Based on the set of privileges, the access right of a file is defined. The exact definition of a privilege varies across applications. For example, we may define a *role-based* privilege [9], [19] according to job positions (e.g., Director, Project Lead, and Engineer), or we may define a *time-based* privilege that specifies a valid time period (e.g., 2014-01-01 to 2014-01-31) 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 covers 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. If the file is a duplicate, then all its blocks must be duplicates as well; otherwise, the user further performs the block-level duplicate check and identifies the unique blocks to be uploaded. Each data copy (i.e., a file or a block) is associated with a token for the duplicate check.

- *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. To reduce the storage cost, the S-CSP eliminates the storage of redundant data via de-duplication and keeps only unique data. In this paper, we assume that S-CSP is always online and has abundant storage capacity and computation power.

- *Data Users.* A user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting de-duplication, 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. In the authorized de-duplication system, each user is issued a set of privileges in the setup of the system. Each file is protected with the convergent encryption key and privilege keys to realize the authorized de-duplication with differential privileges.

- *Private Cloud.* Compared with the traditional de-duplication architecture in cloud computing, this is a new entity introduced for facilitating user's secure usage of cloud service. 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 respectively.

Hybrid clouds generally having twin clouds (private cloud and public cloud). This architecture is used for data de-duplication. For example, an enterprise might use a public cloud service, such as Amazon S3, for archived data, but continue to maintain in-house storage for operational customer data. Alternatively, the trusted private cloud could be a cluster of virtualized cryptographic co-processors, which are offered as a service by a third party and provide the necessary hardware based security features to implement a remote execution environment trusted by the users.

3.2 Adversary Model:

Both the public and private clouds are "honest-but-curious" try to find out as much secret information as possible based on their possessions either within or out of the limits of privileges users would try to access data.

In this one, all the files are sensitive and needed to be fully protected against both public and private cloud. Assume two kinds of adversaries are considered.

- External adversaries which aim to extract secret information as much as possible from both public cloud and private cloud.

- Internal adversaries who aim to obtain more information on the file from the public cloud and duplicate-check token information from the private cloud outside of their scopes.

These adversaries may include S-CSP, private cloud servers and authorized users.

3.3 Design Description:

The detailed architecture of the design is showed in figure2. We can get the processing details from this architecture. Four different types of modules are present in the architecture. Data Owner Module, Encryption and Decryption Module, Remote User Module, Cloud Server Module. User login details are required to upload or download a file and the details of modules mentioned below,

- DATA OWNER MODULE :
  a. Data Owner login validations.
  b. Upload Files.
  c. Manipulates Encrypted files.
  d. Differential Authorization.

- ENCRYPTION AND DECRYPTION MODULE :
  a. Generate signs.

b. Encrypts and uploads files.
c. Decrypts and downloads files.
d. Data confidentiality.

- REMOTE USER MODULE :
  a. Accessing Files.
  b. Remote User login validations.

- CLOUD SERVER MODULE :
  a. Authorized Duplicate Check.
  b. Accessing files.

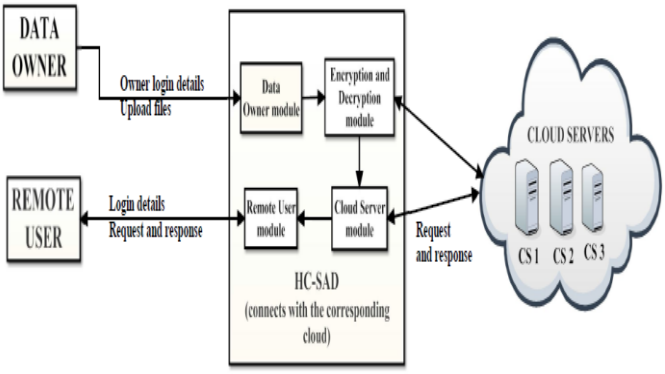

Fig-2. System Architecture design

3.3.1 New deduplication system

In this, we address the problem of privacy preserving de-duplication in cloud computing and propose a new deduplication system supporting for, the

- *Differential Authorization*: To perform duplicate check based on privilege of user is able to get his/her individual token. Without aid from the private cloud server and for the duplicate check outs token cannot generate by the user.

- *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. The security requirements considered in this paper 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 un-forge ability and in-distinguish ability of file token. The details are given below.

- *Unforgeability 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 unforgeability of file tokens. In our 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 our 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. In another word, the goal of the adversary is to retrieve and recover the files that do not belong to them. In our system, compared to the previous definition of data confidentiality based on convergent encryption, a higher level confidentiality is defined and achieved.

4. ALGORITHMS USED:

In this section, we use two types of algorithms,

1). For file uploading.

2). For file downloading.

FOR UPLOADING A FILE

BEGIN

Step –1 Read file

Step –2 Cloud server checks for duplication

Step –3 Sends duplication response whether the file already exists or not

Step – 4 If the file does not exist
4.1 Display “file does not exist”

Step – 5 Then it uploads the file

Step – 6 If the file already exist

6.1 Display "file already exist"

END

FOR DOWNLOADING A FILE

BEGIN

Step –1 Read file

Step –2 Cloud server checks for duplication

Step –3 Sends duplication response whether the file already exists or not

Step –4 If the file exist

-4.1 Display "file exist"

Step –5 then it downloads the file

Step –6 If the file does not exist

-6.1 Display "file does not exist"

END

5. IMPLEMENTATION:

We implement a prototype of the proposed authorized De-duplication system, in which we model three entities as separate C++ programs. 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 key and handles the file token computation. A *Storage Server* program is used to model the S-CSP which stores and de-duplicates files.

We implement cryptographic operations of hashing and encryption with the OpenSSL library [1].We also implement the communication between the entities based on HTTP, using GNU Libmicrohttpd [10] and libcurl [13]. Thus, users can issue HTTP Post requests to the servers.

Our implementation of the Client provides the following function calls to support token generation and de-duplication 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; and
- *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; and
- *ShareTokenGen(Tag, {Priv.})* - It generates the share token with the corresponding privilege keys of the sharing privilege set with HMAC-SHA-1 algorithm.

Our implementation of the Storage Server provides de-duplication 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; and
- *FileStore(FileID, File, Token)* - It stores the File on Disk and updates the Mapping.

6. EXPERIMENTAL RESULTS:

The final results of the designed system are given below. From those results we get the detailed information to Check de-duplication and upload the files, Fetching the Signs using Hashing Algorithm, Checking for Duplication, file uploading, file downloading and attacker trying to attack(block) the cloud. Detailed procedure of the

proposed system is given. Based on this we confirm that securely authorized de-duplication is successfully achieved with hybrid cloud approach. The output images given as below,

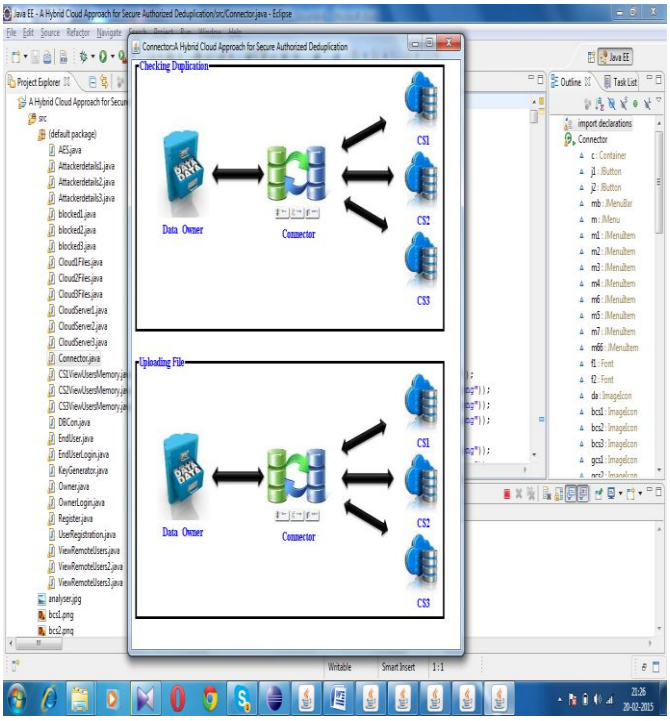

Figure 3. Checking duplication and uploading the files

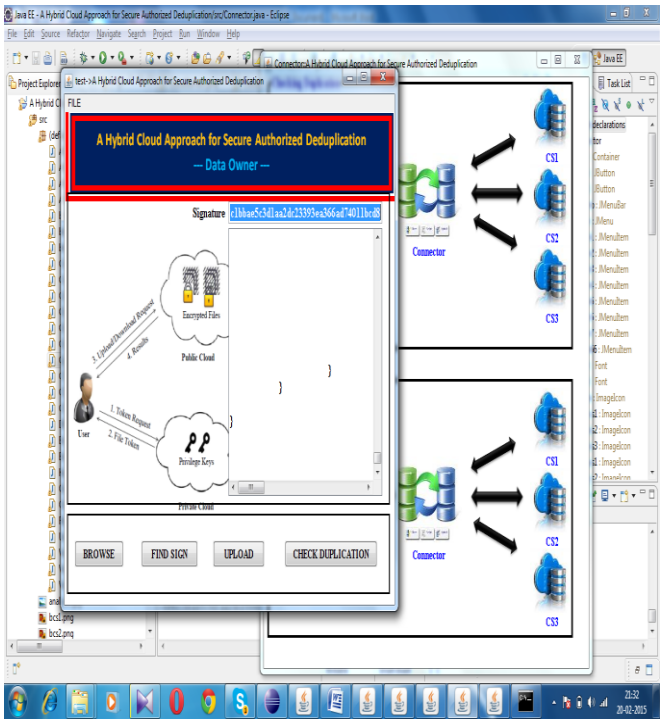

Figure 4. Fetching the Signs using Hashing Algorithm

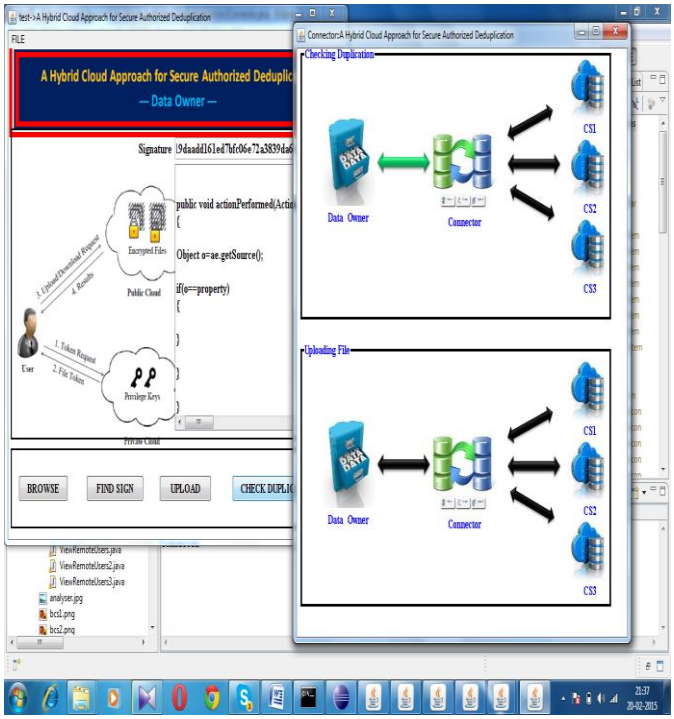

Figure5(a). Checking for Duplication.

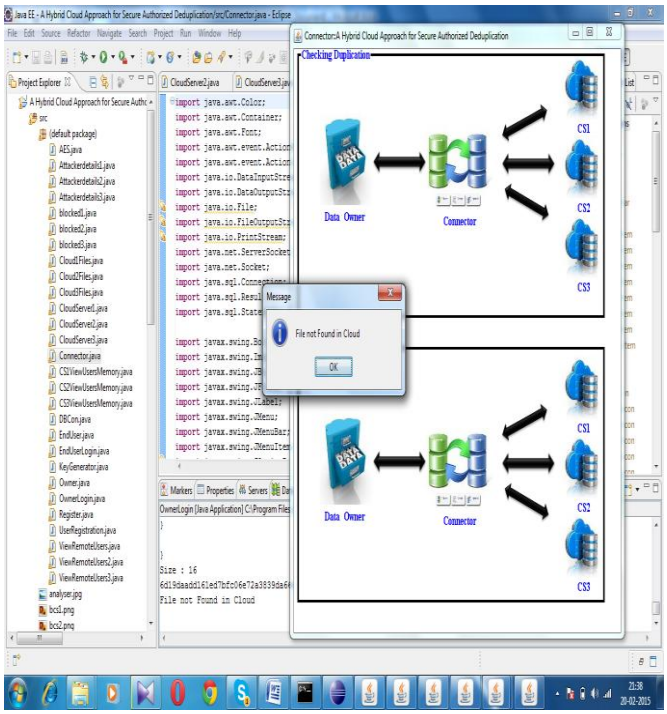

Figure 5(b). File not found in the cloud after checking for duplication

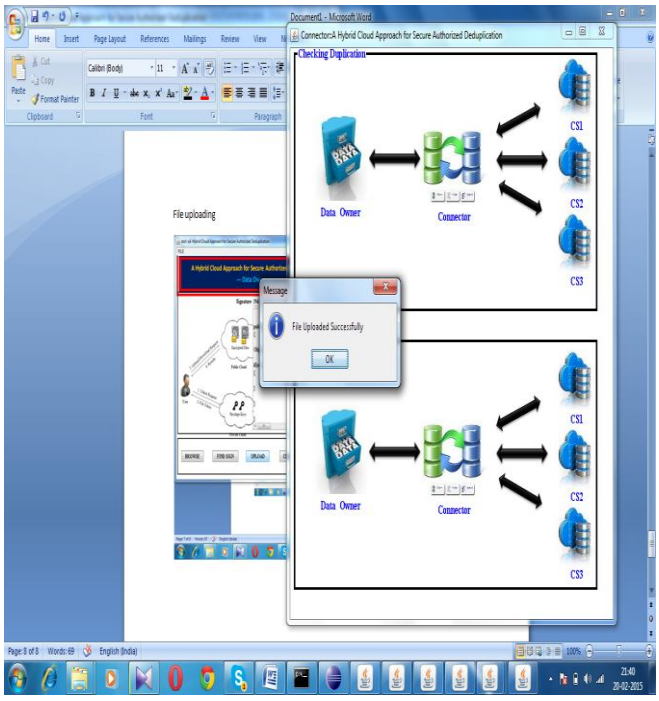

Figure7. Successfully uploaded the file

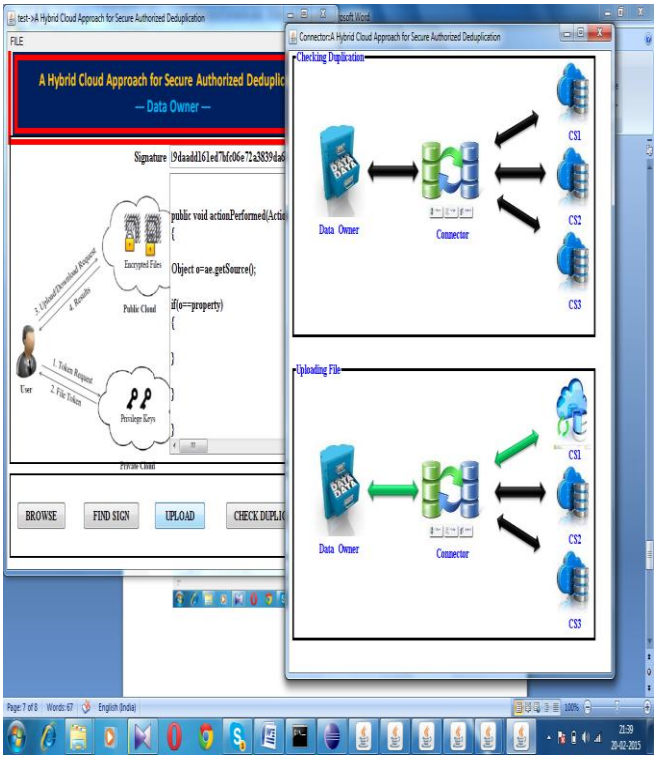

Figure6. File uploading

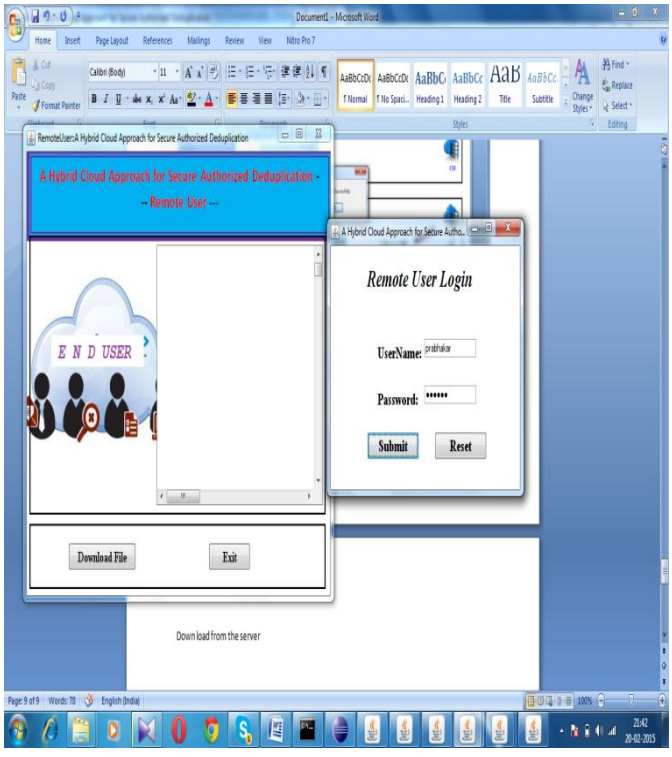

Figure8. Downloading files from the server

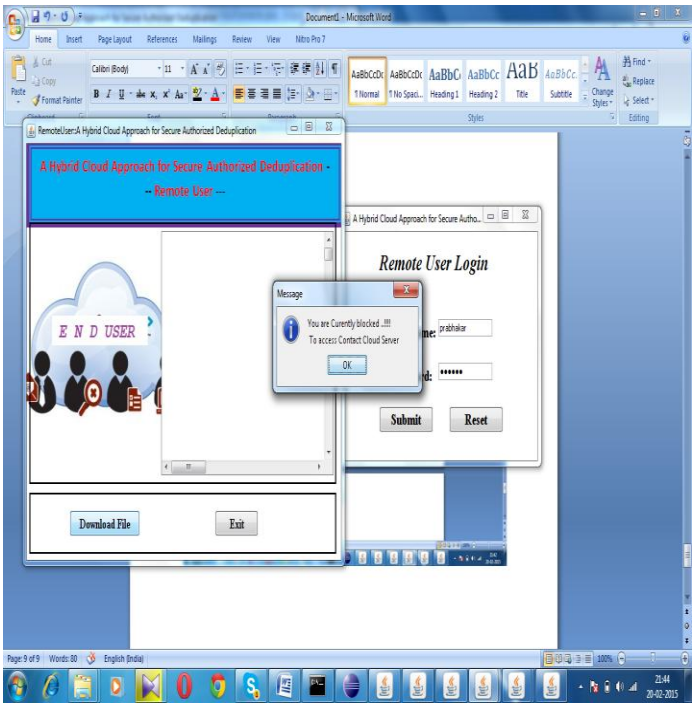

Figure9. Attacker trying to attack the cloud (Blocked)

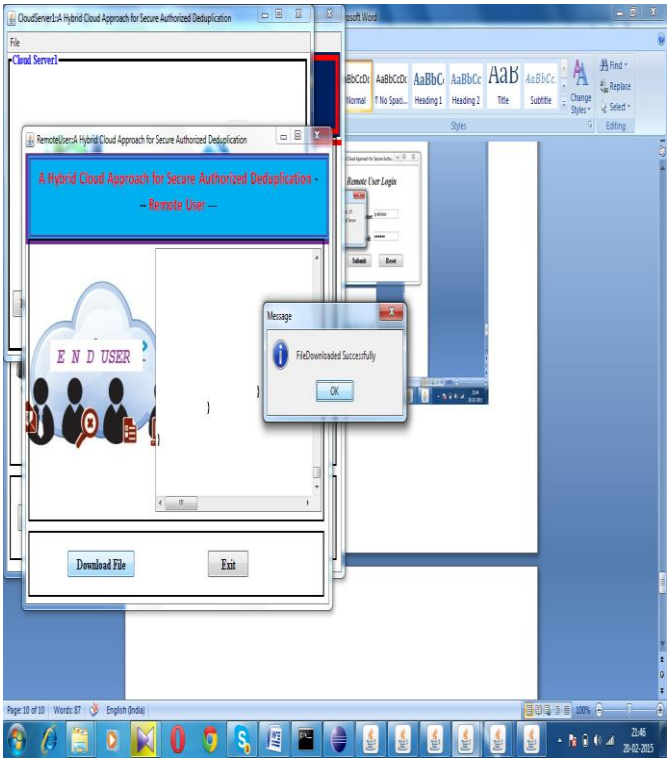

Figure10. Successfully downloading the file if a valid user is logging in

7. CONCLUSION,

Notion of authorized data de-duplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new de-duplication 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. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test-bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

REFERENCES

[1] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou" A Hybrid Cloud Approach for Secure Authorized De-duplication" in vol: pp no-99, IEEE, 2014

[2] OpenSSL Project. http://www.openssl.org/.

[3] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In *Proc. of USENIX LISA*, 2010.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In *USENIX Security Symposium*, 2013.

[5] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure eduplication. In *EUROCRYPT*, pages 296– 312, 2013.

[6] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. *Cryptology*, 22(1):1–61, 2009.

[7] M. Bellare and A. Palacio. Gq and schnorr dentification schemes: Proofs of security against impersonation under active and concurrent attacks. In *CRYPTO*, pages 162–177, 2002.

[8] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. chneider. Twin clouds: An architecture for secure cloud computing. In *Workshop on Cryptography and Security in Clouds (WCSC 2011)*, 2011.

[9] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In *ICDCS*, pages 617–624, 2002.

[10] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15th NIST-NCSC National Computer Security Conf.*, 1992.

[11] GNU Libmicrohttpd. Http://www.gnu.org/software/libmicrohttpd/.

[12] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg. Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 491–500. ACM, 2011.

[13] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure deduplication with efficient and reliable convergent key management. In *IEEE Transactions on Parallel and Distributed Systems*, 2013.

[14] libcurl. http://curl.haxx.se/libcurl/.

[15] C. Ng and P. Lee. Revdedup: A reverse deduplication storage system optimized for reads to latest backups. In *Proc. of APSYS*, Apr 2013.

BIOGRAPHIES

Ms. Jadapalli. Nandini, M.Tech Student, Dept of CSE, Narayana Engineering College, Nellore. Received B.Tech in Computer Science and Engineering from Narayana Engineering College, Nellore. Interesting Areas are Computer Networks and Cloud Computing.

Mr. Ramireddy Navateja Reddy, M.Tech., Assistant Professor, Dept of CSE, Narayana Engineering College, Nellore. Received B.Tech in Information Technology from PBRVITS, Kavali, affiliated to JNTU Hyderabad. in 2007. Received M. Tech degree in Computer Network Engineering from RVCE Bangalore, affiliated to VTU Belgaum, in 2009. Having 6 years of teaching experience. Interesting area is Computer Networks.

