Journal of Computer Science 10 (2): 210-223, 2014
ISSN: 1549-3636
© 2014 Science Publications
doi:10.3844/jcssp.2014.210.223 Published Online 10 (2) 2014 (http://www.thescipub.com/jcs.toc)

DESIGN AND IMPLEMENTATION OF A PRIVACY PRESERVED OFF-PREMISES CLOUD STORAGE

[1]Sarfraz Nawaz Brohi, [1]Mervat Adib Bamiah, [1]Suriayati Chuprat and [2]Jamalul-lail Ab Manan

[1]Advanced Informatics School, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
[2]Strategic Advanced Research Cluster, MIMOS Berhad, Kuala Lumpur, Malaysia

Received 2013-09-21; Revised 2013-10-14; Accepted 2013-11-13

ABSTRACT

Despite several cost-effective and flexible characteristics of cloud computing, some clients are reluctant to adopt this paradigm due to emerging security and privacy concerns. Organization such as Healthcare and Payment Card Industry where confidentiality of information is a vital act, are not assertive to trust the security techniques and privacy policies offered by cloud service providers. Malicious attackers have violated the cloud storages to steal, view, manipulate and tamper client's data. Attacks on cloud storages are extremely challenging to detect and mitigate. In order to formulate privacy preserved cloud storage, in this research paper, we propose an improved technique that consists of five contributions such as Resilient role-based access control mechanism, Partial homomorphic cryptography, metadata generation and sound steganography, Efficient third-party auditing service, Data backup and recovery process. We implemented these components using Java Enterprise Edition with Glassfish Server. Finally we evaluated our proposed technique by penetration testing and the results showed that client's data is intact and protected from malicious attackers.

Keywords: Cloud Storage, Privacy, Security, Confidentiality, Integrity

1. INTRODUCTION

Cloud computing offers an innovative method of delivering computing resources whereby clients are able to execute their applications at remote servers with unlimited storage capability while enjoying efficient features such as scalability, availability, on-demand self-service and elasticity on pay-per-use billing pattern (Yashpalsinh and Modi, 2012). The cloud computing paradigm has brought up an agile and revolutionized IT infrastructure for business organizations, since they can now focus on their core business while transferring the IT responsibilities such as managing servers, storages, developing applications and installing networks, to a Cloud Service Provider (CSP) (Khayyam *et al.*, 2012). Due to cost-effective operational efficiency, business organizations are rapidly adopting the cloud paradigm. However migrating data to the cloud is still a serious concern for the organizations requiring consistent confidentiality and integrity (Rocha and Correia, 2011).

In order to formulate privacy preserved off-premises cloud storage, in this research paper, we propose an improved technique that will give confidence to users in using the cloud storage for their daily transactions via cloud applications. Our proposed privacy preserved off-premises cloud computing storage solution may not be appealing enough to ordinary users who do not need strict security as well as privacy requirements. Ordinary users also do not deal with critical data or follow any industry based regulatory compliance standards such as Health Insurance Portability and Accountability Act (HIPAA) or Payment Card Industry Data Security Standard (PCIDSS).

Since proposed solution requires client to perform number of tasks such as encryption, decryption, metadata generation, requesting and inserting security code for each transaction, ordinary users may consider the overall computing process as time consuming. However, to protect the sensitive information of an organization, these tasks are quite essential. For-instance nowadays in

Corresponding Author: Sarfraz Nawaz Brohi, Advanced Informatics School, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

online banking systems, users are required to complete a set of security requirement to perform an operation, they follow these processes appropriately to protect their confidential information from malicious attacks, so there has to be a trade-off among security, privacy and performance.

2. ORGANIZATIONS AND CLOUD SECURITY

Organizations that are required to follow well-defined data security standards such as HIPAA and PCIDSS do not trust the existing security as well as privacy policies offered by CSPs (Mervat *et al.*, 2012; Shucheng *et al.*, 2010). They feel the lack of control while storing confidential records at off-premises storage and they are concerned that malicious users might gain illegal access to their records. In order to defend the attacks, CSPs formulate their cloud infrastructure with efficient security controls such as network firewalls, secure cryptographic processors, anti-malwares, honey pots, access control mechanisms. However cloud storages using these techniques are also subjected to vulnerabilities in terms of implementation (Rocha and Correia, 2011).

Clients require full confidence that attackers are not able to steal, view or tamper their data. We know from the past studies that security and privacy on cloud is breached by external or internal attackers (Ling *et al.*, 2011). External attacks are issued by hackers who steal client's confidential records with an objective to obtain desired amount of cash. These attacks may take place by an IT personal belonging to competitors of CSP or client. The intention of these attacks is to damage the brand reputation of CSP or to abuse as well as misuse client's files. CSPs secure their physical and virtual infrastructure by using various tools and techniques to protect data and their systems from outsider attacks. However, we found out that existing solutions are not adequate to preserve the client's privacy. It is also identified that internal employees of CSP may become malicious as well (Catteddu and Hogben, 2009).

Inside attackers such as malicious employees of CSPs, intentionally exceed their privileged accesses in a negative manner to affect the confidentiality (Adrian *et al.*, 2012). In contrast to an external hacker, malicious insider can attack the computing infrastructure with relatively easy manner and less knowledge of hacking, since they have the detailed description of underlying infrastructure. CSPs admitted their full awareness aboutpossible malicious insiders and they would normally claim to have its solution.

For-example, in order to hire a trusted cloud admin, CSPs conduct strict background checks and multiple detailed interviews whereas some CSPs state that they have strict security procedures in place for all employees when they have access to machines that store client's files. Although these are essential security steps that they must follow, but without acomplete trustworthy solution for defending insider attacks, a malicious insider can easily obtain passwords, cryptographic keys, files and gain access to client's records (Francisco *et al.*, 2011). When client's data confidentiality has been breached, client would never have any knowledge of the unauthorized access mostly due to lack of control and transparency in cloud provider's security practices and policies.

2.1. Related Work

Ateniese *et al.* (2007) proposed Provable Data Possession (PDP) model to ensure that external storage retains the client's file with required integrity policies. Using the PDP model, data owner i.e. the client first pre-processes the file F and adds some additional data or expands it such as the new file is F', client then generates Verification Metadata (VMd) denoted as M for F' before sending it to the server for storage. Generated M will be stored locally and client may delete the personal copy of F, but prior to the deletion, client will execute a data possession challenge to make sure that server has successfully retained the file. The yes or no response from server will verify the existence of file. While uploading the file, client may encrypt it or it may already be encrypted. To verify the file integrity, client issues a challenge and request R to server to compute hash for the stored file. Server will compute hash and send the result P to the client and client will verify the integrity results by comparing it to locally stored metadata M.

Wang *et al.* (2010) introduced an improved technique of verifying data integrity on cloud by utilizing concept of Third Party Auditor (TPA). Since client doesnot have knowledge and expertise of auditing process, TPA will conduct auditing services on behalf of client. TPA is a certified and trusted entity for conducting scheduled or requested auditing. This technique of using TPA, enables the client to send the file by dividing it into number of blocks, so file F will be denoted as sequence of N blocks i.e., m_1,…,mi,…,mn. For each block Message Authentication Code (MAC) will be generated by using the following function:

$$K \times \{0,1\}^* \rightarrow \{0,1\}^1$$

where, K denotes the key space.

Using the TPA concept, the cloud user first generates the public and private key parameters and then signs each block and sends file with its VMd to server. As TPA receives the file block number and verifies its signature, quits by indicating false if verification has failed or process continues if the result is true.

Venkatesh *et al*. (2012) provided a solution similar to (Ateniese *et al*., 2007; Wang *et al.*, 2010) but enhancing them by adding Rivest Shamir Adleman (RSA) algorithm based storage security. Client generates signature for each block using RSA secret key and hashing algorithm Ti = (H(mi). gmi rsk and generates signature Φ = {Ti} for collection of all blocks. Merklee Hash Tree (MHT) is constructed by involving each block. Client then signs the root of MHT with secret key as sigrsk(H(R)) = H(R)rsk. In next step client sends the file, its hash and signed root {F,Φ, sigrsk(H(R))} to server and deletes the local copy of Φ and sigrsk(H(R)). During the auditing process, client issues a challenge to server by selecting a specific file block, while the server computes the proof and sends the results back to client. Client verifies the data integrity by using the MHT root and authenticates by using the secret key.

Ranchal *et al.* (2010) argued that the concept of involving TPA in cloud services may associate additional risk to confidentiality. TPA may not act as expected (this happen when TPA is attacked by external hacker or malicious insider) and they counter a solution for protecting information integrity in cloud without using TPA. They proposed an Identity Management (IDM) approach using active bundle scheme which includes sensitive data, metadata and a virtual machine. In order to maintain confidentiality, the Personally Identified Information (PII) is packed and encrypted inside the active bundle. Virtual machine monitors and manages the program inside the bundle that actually controls the access to bundle.

Prasadreddy *et al.* (2011) focused on key and data management technique while preserving the privacy of users on cloud. They proposed a web-browser plug-in that enables the client to store the key and data with isolated CSPs. In this architecture, it is not required to have mutual communication between two CSPs, rather the plug-in handles all the necessary tasks. For-example if file is stored with Amazon, keys will be stored with Google, so Amazon cannot have illegal access to file since it is obfuscated and it can only be de-obfuscated using the corresponding key. Each file and its key will have a unique tag. If user wants to access the file, the data storing CSP sends the obfuscated data with its associated tag to the plug-in and then plug-in sends request to the key storing CSP by providing the tag. The key storing CSP will check the tag, identify the corresponding key and sends it to the plug-in and finally plug-in de-obfuscates the data by using that secret key.

3. A PRIVACY PRESERVED OFF-PREMISES CLOUD STORAGE

Considering the requirements, we proposed a privacy preserving technique that enables an organization dealing with sensitive information to store their confidential data at off-premises cloud storage without security and privacy concerns. We have designed as well implemented an improved technique which focuses on achieving following set of requirements:

- Only privileged users can access the system under a strict access control policy
- Grant adequate control to client on his data, such as handling encryption, decryption and VMd generation tasks
- Client can process the data and perform certain transactions without decryption
- Client can consistently monitor his files on cloud without revealing any data to an illegal authority
- Client can efficiently restore the unintentionally written, modified or violated records

3.1. System Architecture

The proposed system architecture consists of three end-users i.e., Client, Trusted-TPA (TTPA) and Cloud admin. From client’s perspective, we have assumed that cloud admin is most un-trusted, TTPA is semi-trusted and client’s personal admin is a fully trusted user. This is based on the assumption that TTPA may have been hijacked by a malicious attacker. In order to ensure a safe computing environment, each involved entity is only permitted to perform his privileged tasks but with added security implementation of Role-based Access Control (RBAC) with random Security Code Generator (SCG). Cloud server is the central component of overall system. It analyzes the access control mechanism and performs corresponding operations requested by the privileged users as shown in **Fig. 1**.

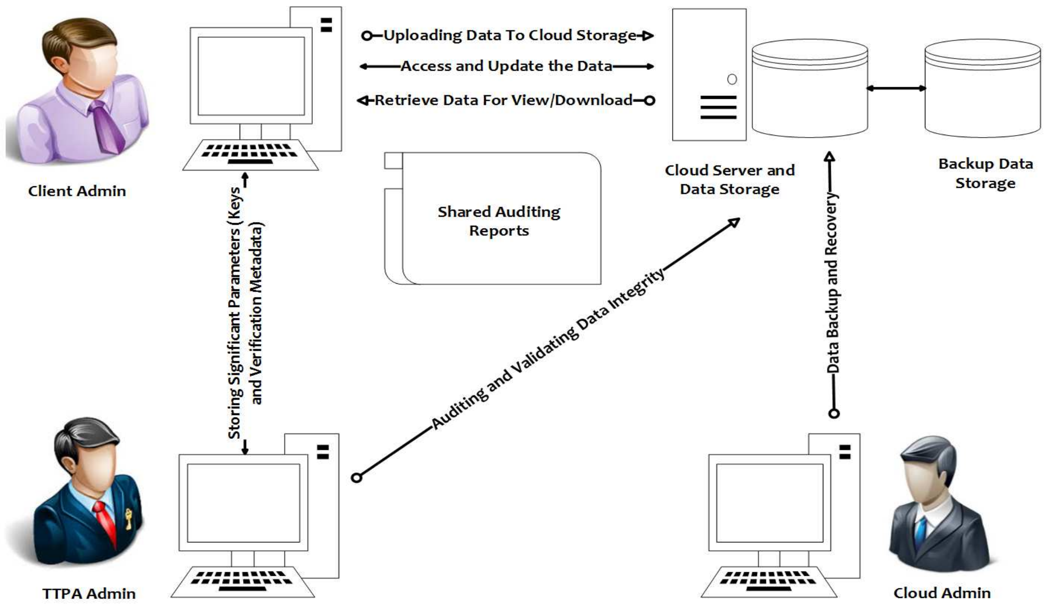

Client Admin
Uploading Data To Cloud Storage
Access and Update the Data
Retrieve Data For View/Download
Cloud Server and Data Storage
Backup Data Storage
Shared Auditing Reports
Storing Significant Parameters (Keys and Verification Metadata)
Auditing and Validating Data Integrity
Data Backup and Recovery
TTPA Admin
Cloud Admin

Fig. 1. System architecture

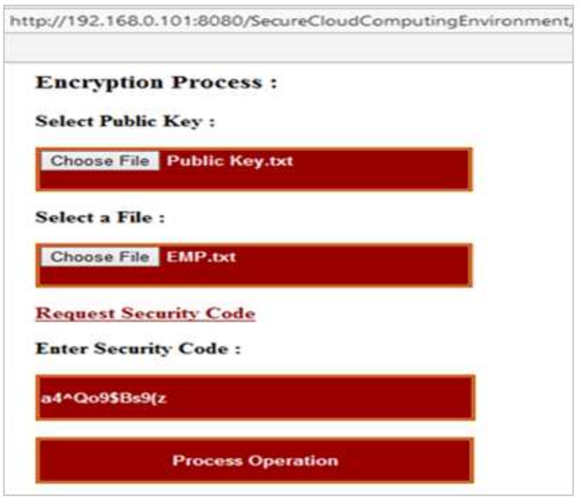
For maintaining data confidentiality and integrity, the responsibilities of client admin are as follows:

- Homomorphic encryption of data prior to its arrival at the cloud storage
- Decrypt, update and download data throughout the entire computing life cycle
- Safe delivery of significant parameters such as hashes, private and public keys
- Ensure with the assistance of TTPA admin that data on cloud is intact

TTPA admin belongs to an auditing authority and is responsible for following tasks:

- Conducting the auditing services on behalf of client
- Initiating scheduled or requested auditing process as directed by client admin
- Providing response to the client about the status of data,whether it is tampered, deleted, manipulated or intact
- Requesting the cloud admin to start data recovery process from backup cloud storage
- Storing the client's parameters such as keys and VMd

The auditing reports will be shared among all three involved parties to determine the integrity status. Cloud admin is responsible for successful recovery of certain amount of records that has been violated by authorized or unauthorized users.


http://192.168.0.101:8080/SecureCloudComputingEnvironment,

Encryption Process :

Select Public Key :

Choose File Public Key.txt

Select a File :

Choose File EMP.txt

Request Security Code

Enter Security Code :

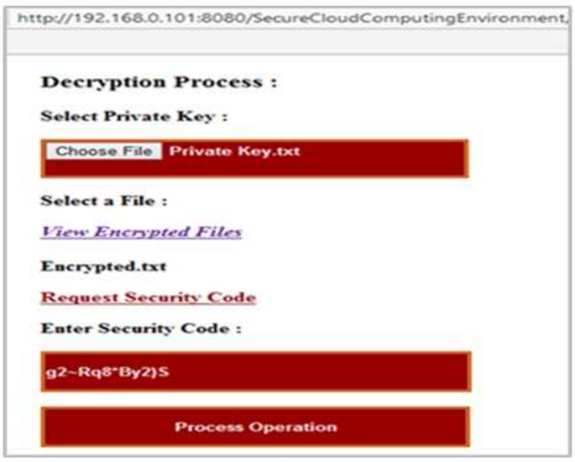
a4^Qo9$Bs9(z

Process Operation

Fig. 2. Encryption

3.2. System Workflow

The process is initiated from client admin with the generation of private and public keys by requesting the cloud server. Let us examine a simple scenario. For-instance client admin wants to store a file named as *EMP.txt* containing organization's employees' confidential records at the cloud storage. Cloud server requires the file and the public key for encryption process as represented in **Fig. 2**.


http://192.168.0.101:8080/SecureCloudComputingEnvironment

Decryption Process :

Select Private Key :

Choose File Private Key.txt

Select a File :

View Encrypted Files

Encrypted.txt

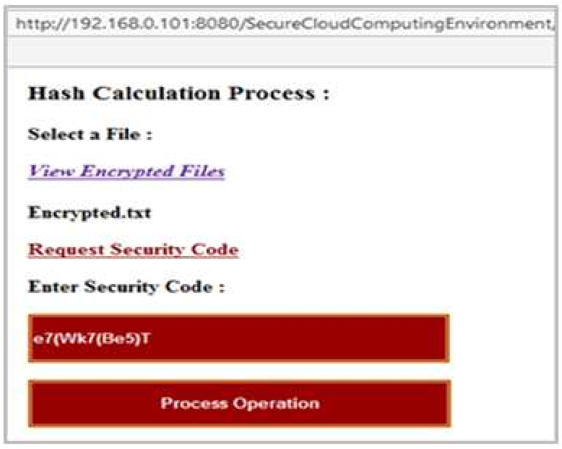
Request Security Code

Enter Security Code :

g2~Rq8*By2)S

Process Operation

Fig. 3. Decryption


http://192.168.0.101:8080/SecureCloudComputingEnvironment

Hash Calculation Process :

Select a File :

View Encrypted Files

Encrypted.txt

Request Security Code

Enter Security Code :

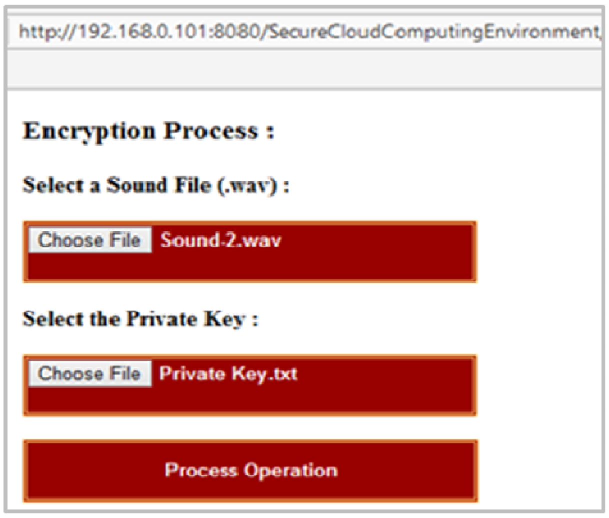
e7(Wk7(Be5)T

Process Operation

Fig. 4. Metadata generation

Data inside the file will be homomorphically encrypted from the stream during the uploading process and when file arrives at the server it will be completely encrypted and stored. File *EMP.txt* is renamed as *Encrypted.txt* when it is saved at the cloud storage. Whenever required, server needs client's private key to decrypt the selected file as shown in **Fig. 3**.

Each attribute is retrieved from storage, decrypted and presented for client's view. Since data is homomorphically encrypted, client admin can also perform certain number of transactions without actually decrypting the contents. Client admin requests the server to compute VMd for the file. Cloud server divides the file in equal number of blocks. In this example, *Encrypted.txt* is divided into four blocks according to its size and server generates metadata for the entire file as well as its each block as shown in **Fig. 4**.


http://192.168.0.101:8080/SecureCloudComputingEnvironment

Encryption Process :

Select a Sound File (.wav) :

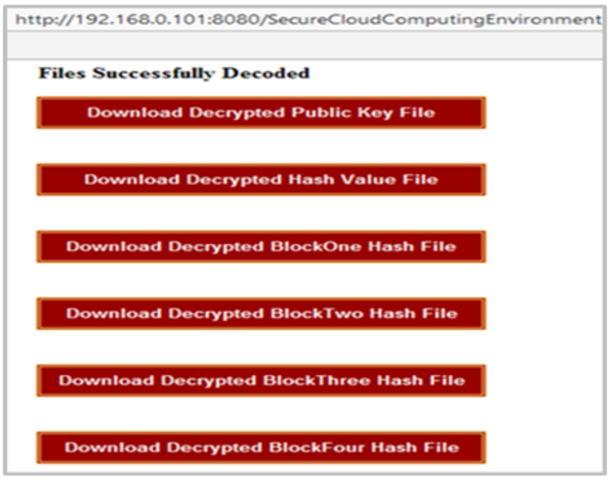
Choose File Sound-2.wav

Select the Private Key :

Choose File Private Key.txt

Process Operation

Fig. 5. Sound steganography


http://192.168.0.101:8080/SecureCloudComputingEnvironment

Files Successfully Decoded

Download Decrypted Public Key File

Download Decrypted Hash Value File

Download Decrypted BlockOne Hash File

Download Decrypted BlockTwo Hash File

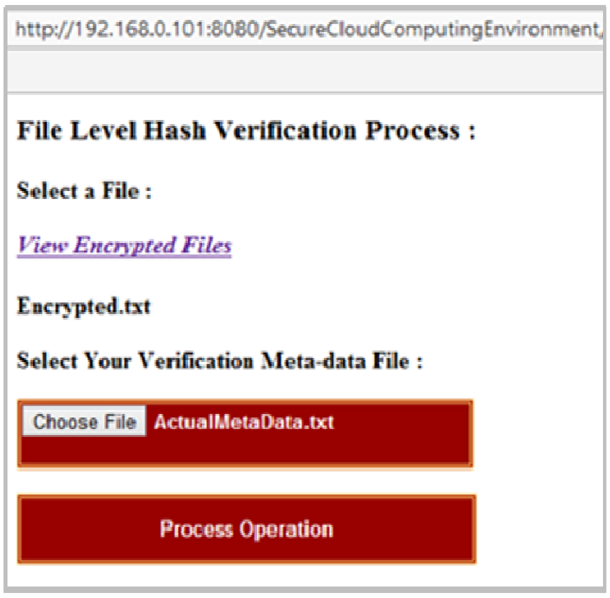
Download Decrypted BlockThree Hash File

Download Decrypted BlockFour Hash File

Fig. 6. Decoded sound-1

Client admin downloads and store the VMd at his local storage and request the cloud server to encrypt the public key, private key and VMd. Cloud server encrypts these parameters using sound steganography as shown in **Fig. 5** and transfers them to TTPA admin as two isolated sound files i.e., sound-1 containing the public key and VMd, sound-2 containing the private key.

At this stage, client admin proceeds with the deletion of cryptographic keys and VMd from the local storage and request the TTPA admin to initiate the auditing process. In order to audit the client's files, TTPA admin needs to extract public key and VMd from the sound-1 by requesting the cloud server as shown in **Fig. 6**.



http://192.168.0.101:8080/SecureCloudComputingEnvironment/

File Level Hash Verification Process :

Select a File :

View Encrypted Files

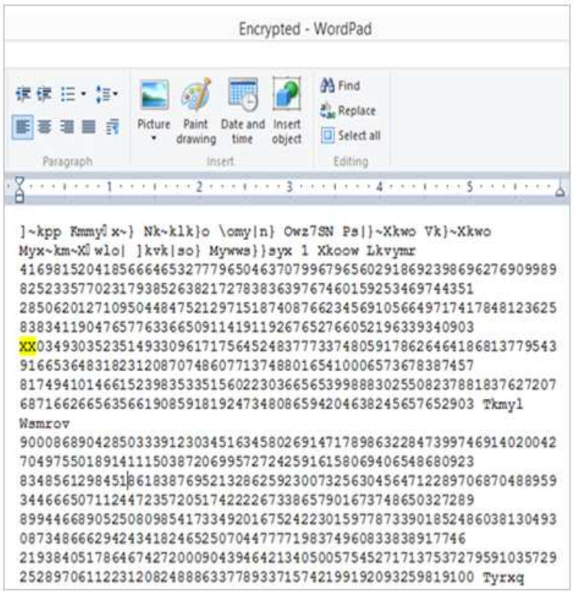
Encrypted.txt

Select Your Verification Meta-data File :

Choose File ActualMetaData.txt

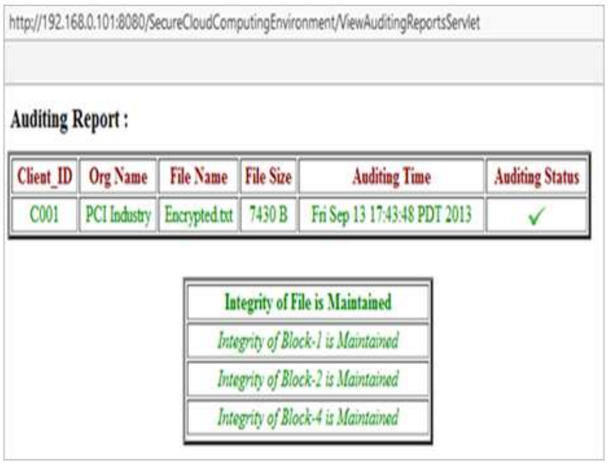
Process Operation

Fig. 7. Auditing process



```
]~kpp Kmmy x~} Nk~klk}o \omy|n} Owz7SN Ps|}~Xkwo Vk}~Xkwo
Myx~km~X wlo| ]kvk|so} Mywws}}syx 1 Xkoow Lkvymr
41698152041856664653277796504637079967965602918692398696276909989
8252335770231793852638217278383639767460159253469744351
28506201271095044847521297151874087662345691056649717417848123625
8383411904765776336650911419119267652766052196339340903
XX034930352351493309617175645248377733748059178626464186813779543
9166536483182312087074860771374880165410006573678387457
81749410146615239835335156022303665653998883025508237881837627207
6871662665635661908591819247348086594204638245657652903 Tkmyl
Wsmrov
90008689042850333912303451634580269147178986322847399746914020042
7049755018914111503872069957272425916158069406548680923
83485612984518618387695213286259230073256304564712289706870488959
3446665071124472357205174222267338657901673748650327289
89944668905250809854173349201675242230159778733901852486038130493
087348666294243418246525070447777198374960833838917746
21938405178646742720009043946421340500575452717137537279591035729
2528970611223120824888633778933715742199192093259819100 Tyrxq
```

Fig. 9. Data violation



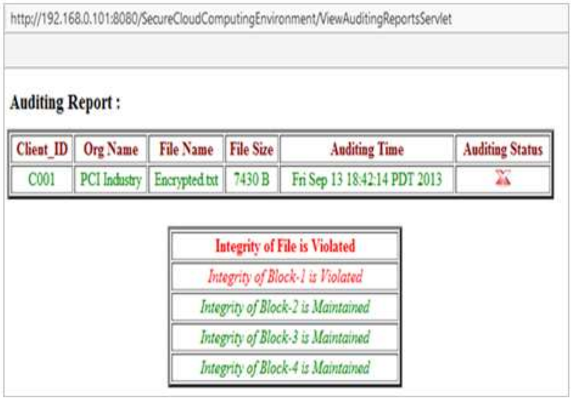
http://192.168.0.101:8080/SecureCloudComputingEnvironment/ViewAuditingReportsServlet

Auditing Report :

Client_ID	Org Name	File Name	File Size	Auditing Time	Auditing Status
C001	PCI Industry	Encrypted.txt	7430 B	Fri Sep 13 17:43:48 PDT 2013	✓

Integrity of File is Maintained
Integrity of Block-1 is Maintained
Integrity of Block-2 is Maintained
Integrity of Block-4 is Maintained

Fig. 8. Initial auditing report



http://192.168.0.101:8080/SecureCloudComputingEnvironment/ViewAuditingReportsServlet

Auditing Report :

Client_ID	Org Name	File Name	File Size	Auditing Time	Auditing Status
C001	PCI Industry	Encrypted.txt	7430 B	Fri Sep 13 18:42:14 PDT 2013	✗

Integrity of File is Violated
Integrity of Block-1 is Violated
Integrity of Block-2 is Maintained
Integrity of Block-3 is Maintained
Integrity of Block-4 is Maintained

Fig. 10. Auditing report after integrity violation

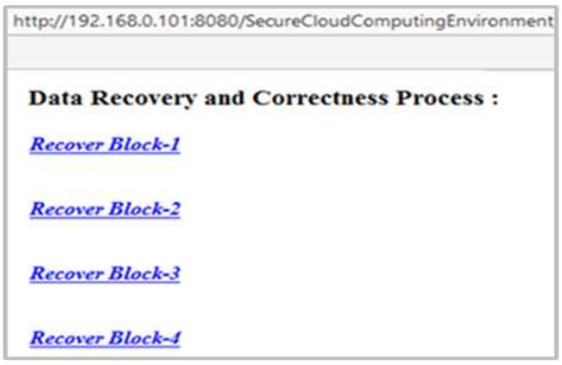
TTPA admin downloads these parameters to his local storage and proceed with auditing process by providing them to the cloud server. Fresh metadata for the stored file is computed by the cloud server and compared with the old metadata as shown in **Fig. 7**.

Based on the auditing results, cloud server creates the auditing reports and shares it among involved users. Since the file is securely saved with required integrity, auditing reports indicated that integrity of file and each block is well maintained as shown in **Fig. 8**.

However, if file integrity is violated by any user, for-example **Fig. 9** represents that malicious users have modified the first block of file by replacing an encrypted record with garbage characters such as “xx”.

The auditing report clearly indicates the integrity violation and its location i.e. block-1 as shown in **Fig. 10**. TTPA admin requests the cloud admin to overcome the violation and recover the data back to its original state.

Upon receiving the request cloud admin views the auditing reports and identifies that block-1 is violated. Cloud admin will issue a request to cloud server for recovering the block-1 as shown in **Fig. 11**.



http://192.168.0.101:8080/SecureCloudComputingEnvironment

Data Recovery and Correctness Process :

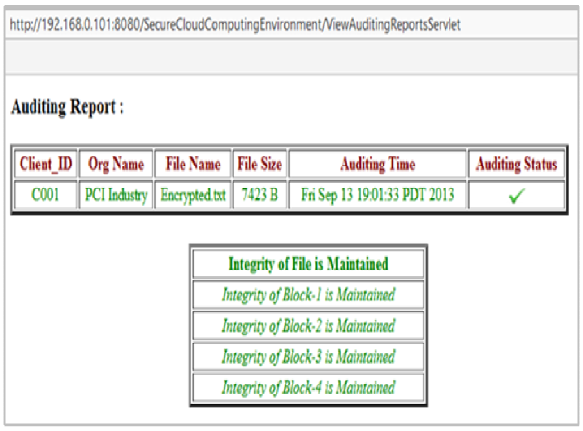
Recover Block-1

Recover Block-2

Recover Block-3

Recover Block-4

Fig. 11. Data recovery process



http://192.168.0.101:8080/SecureCloudComputingEnvironment/ViewAuditingReportsServlet

Auditing Report :

Client_ID	Org Name	File Name	File Size	Auditing Time	Auditing Status
C001	PCI Industry	Encrypted.txt	7423 B	Fri Sep 13 19:01:33 PDT 2013	✓

Integrity of File is Maintained
Integrity of Block-1 is Maintained
Integrity of Block-2 is Maintained
Integrity of Block-3 is Maintained
Integrity of Block-4 is Maintained

Fig. 12. auditing report after data recovery

After the successful recovery, he will alert the TTPA admin to restart the auditing process in order to ensure that data is intact. After the successful recovery, cloud admin will alert the TTPA admin to restart the auditing process in order to ensure that data is intact. The auditing reports after recovery indicate that integrity in well maintained as shown in **Fig. 12** and there is no data loss. At this point, TTPA admin will inform the client admin to proceed with other operations such as uploading more files, update existing records or download files permanently.

Since client admin has already deleted all the parameters from local storage, client will retrieve both sound files from TTPA admin and request the cloud server to extract the public and private key from sound-1 and sound-2 respectively.

3.3. System Implementation

We have described the implementation mechanism of our suggested contributions such as Resilient RBAC mechanism, Partial homomorphic cryptography, Metadata generation and sound steganography, efficient third-party auditing service, Data backup and recovery process. The codes snippets used throughout this paper are written in Java and compiled using NetBeans 6.9.1 runtime environment with Glassfish Server version-3.0.

3.3.1. Resilient Role-based Access Control Mechanism

According to security recommendations provided by Cloud Security Alliance (CSA), data owner is responsible for enforcing access control policies whereas CSP is responsible for their implementation (CSA, 2011). We assumed that data owner together with its IT professionals and legal law authorities will specify and enforce the access control policies by signing a Service Level Agreement (SLA) with CSP. We have implemented the access policies of client by using RBAC.

Three major roles were created for accessing the system i.e., Client, TTPA and Cloud Admin by using Glassfish Server file realm security domain. The access control to perform a particular operation by a privileged user is further implemented using Enterprise Java Beans and security annotations. We assume that client will set the following access privileges for each involved user as shown in **Fig. 13**. Operations such as decryption and encryption of client's data are only associated to client admin, data recovery process is associated to cloud admin and conducting the auditing service is associated to TTPA admin, whereas viewing audit reports is associated to all three involved roles, these accesses are controlled using following code snippet:

```
@RolesAllowed("Client Admin")
public String decryptDataFiles() throws Exception {}
public String encryptDataFiles() throws Exception {}
@RolesAllowed("Cloud Admin")
public String  dataRcoveryProcess() throws Exception {}
@RolesAllowed("TTPA Admin")
public String initiateAuditingProcess() throws
Exception{}
@RolesAllowed({"Client Admin","TTPA
Admin","Cloud Admin"})
public String viewAuditingReports() throws Exception
{}
```

In order to further enhance the security of access control, we have implemented a SCG process that generates a random twelve digit code which consists of special character, symbols, numbers, upper and lower case letters.

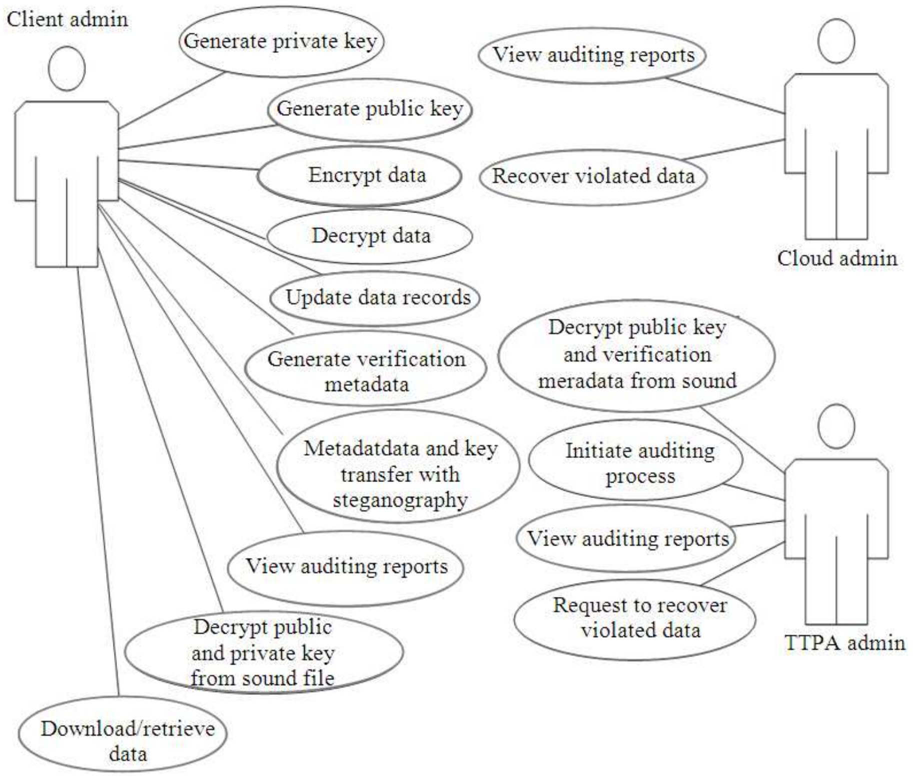

Client admin
Generate private key
Generate public key
Encrypt data
Decrypt data
Update data records
Generate verification metadata
Metadatdata and key transfer with steganography
View auditing reports
Decrypt public and private key from sound file
Download/retrieve data
View auditing reports
Recover violated data
Cloud admin
Decrypt public key and verification meradata from sound
Initiate auditing process
View auditing reports
Request to recover violated data
TTPA admin

Fig. 13. Role-based access control privileges

The SCG was implemented using following code snippet:

```
Random  r=new Random();
String
alphabet[]={"a","b","c","d","e","f","g","h","i","j","k","l"
+
"","m","n","o","p","q","r","s","t","u","v","w","x","y","z"
};
String symbol
[]={"~","!","@","$","%","^","&","*","(",")","?"};
String
alphabetTwo[]={"A","B","C","D","E","F","G","H","I","
J","K","L"
+
"","M","N","O","P","Q","R","S","T","U","V","W","X","
Y","Z"};
String alpha;
int number;
int c=6;
int c2=c/2;
int nc2=0-c2;
String finalCode="";
String alphaTwo;
String alphaThree;
for(int x=nc2;x<0;x++){
  int alphaNum=r.nextInt(26);
  alpha=alphabet[alphaNum];
  number=r.nextInt(10);
  String numStr=Integer.toString(number);
  int symbolNum=r.nextInt(11);
  alphaTwo=symbol[symbolNum];
  int alphaNumTwo=r.nextInt(26);
  alphaThree=alphabetTwo[alphaNumTwo];

finalCode=finalCode+alpha+numStr+alphaTwo+alphaT
hree;
}
```

All transaction such as decryption or auditing cannot proceed unless accurate security code has been requested and provided by the authorized user. Once the code is requested, it will be generated and sent via email or SMS to the appropriate user to proceed with the transaction. On the successful entry of security code, system will perform the operation or else user needs to request a fresh code.

3.3.2. Partial Homomorphic Cryptography

One of the limitations of cloud systems is the enablement of the client to process the data on cloud while it remains encrypted (Hibo *et al*., 2011). In order to perform this task, client is required to decrypt the data prior to processing. We have implemented partial homomorphic cryptography which enables the client to perform certain number of operations on encrypted data. However, it is not possible to perform unlimited number of operations, due to unavailability of practical implementation for full homomorphic cryptography. Considering the security features of asymmetric algorithms, we implemented the homomorphic version of RSA algorithm to encrypt, decrypt and process the encrypted data on cloud. The random homomorphic public and private keys are generated by using following code snippet:

```
BigInteger p=BigInteger.probablePrime(N/2,random);
BigInteger q=BigInteger.probablePrime(N/2,random);
n=p.multiply(q);
BigInteger
phi_n=(p.subtract(one)).multiply(q.subtract(one));
publicKey=new BigInteger("65537");
privateKey=publicKey.modInverse(phi_n);
// where N refers to bit-length of returned BigInteger.
```

The public and private keys are generated and stored with user. Using the public key client admin can then start the encryption by using the following code which is multiplicative homomorphic.

```
message.modPow(publicKey,n);
// where message refers to the data being encrypted.
```

Similarly, data will be decrypted using the following code:

```
encrypted.modPow(privateKey,n);
// where encrypted refers to the cipher data.
```

Since the data is encrypted by using RSA multiplicative homomorphism, client processes the encrypted data by performing number of transaction that are multiplicative in nature. For-example when file *EMP.txt* is stored at the cloud, client can increase, modify or delete the salary and commission rate of an employee. Some demo queries are implemented as follows:

Query-1: Increase the commission rate of an employee.

```
inputFactor=2;
encrypt(inputFactor,publickKey).
inputFactor.muptiply(commission_rate).
```

Query-2: Alter and update the salary of an employee to 1350.

```
inputFactor=1350;
multiplicativeFactor=1;
encrypt(inputFactor,publickKey).
encrypt(multiplicativeFactor,publickKey).
amount=inputFactor.mupltiply(multiplicativeFactor).
salary=amount.
```

Similarly, there can be various other operations that can be performed on encrypted records.

3.3.3. Metadata Generation and Sound Steganography

When data is stored at the cloud, client admin needs to generate VMd by using file and block level hash calculation methods. Client first generates the VMd for entire file and then for its each block. The file will be automatically divided by server in N number of blocks according to its size. For-example if the file F is 8MB, it will be divided in four equal chunks F1, F2, F3 and F4 each of size 2MB. VMd for entire file will be calculated by using Digital Signature Algorithm (DSA) with SHA-1, as represented in following code snippet:

```
Signature
dsa=Signature.getInstance("SHA1withDSA","SUN");
dsa.initSign(privateKey);
dsa.update(fileData);
verificationMetadata=dsa.sign();
```

Similarly VMd for each block will be generated, but instead of file, block number will be provided as parameter i.e. dsa.update(fileBlock_n), where n refers to a specific block number. After metadata generation process is accomplished, the client admin will send the VMd and public key to TTPA admin for conducting the auditing process. Client admin will also send his private key for secure storage. However, to verify the data integrity TTPA admin only requires the VMd and client's public key. In order to protect these parameters from man-in-middle and other malicious attacks, parameters such as keys and VMd are first encoded using Base64 encoding scheme for the transmission over the network and secondly by using sound steganography techniques, as represented in following code snippet:

```
BASE64Encoder encoder=new BASE64Encoder();
String message=encoder.encode(parameter);
// Sound steganography.
Sound s=new Sound(inputFile);
int nbrOfSamples=message.length()*3;
```

```
// where message refers to the value being encoded.
smoothAmpValues(s,nbrOfSamples+3);
for(int g=0;g<nbrOfSamples;g++) {
  int asciiValue=(int) message.charAt(g/3);
  int digit=getDecimalDigit(asciiValue,g%3);
  int ampValue;
  if(s.getSampleValueAt(g)>=0) {
    ampValue=s.getSampleValueAt(g)+digit;
  }
 else{
   ampValue=s.getSampleValueAt(g)-digit;
 }
if(ampValue>32767){
  ampValue=ampValue-10;
 } if (ampValue<-32768){
  ampValue=ampValue+10;
 }
s.setSampleValueAt(g,ampValue);
```

With the completion of steganography process, two encoded sound files will be created, sound-1 contains the public key and VMd and sound-2 contains the private key. TTPA admin is only privileged to decrypt the sound-1, since private key belongs to the client admin and it is just sent for secure storage.

3.3.4. Efficient Third-Party Auditing Service

In order to conduct the auditing process, TTPA admin will request the cloud server to generate fresh VMd for client's file stored at cloud. However, server requires client admin's public key and VMd from TTPA admin to generate the auditing reports. Since parameters sent by client are encoded using sound steganography and Base64, all parameters will be decrypted using the following code snippet:

```
Sound s=new Sound(filename);
SoundSample[] samples=s.getSamples();
int[] digit=new int[10];
String message="";
// where message contains the desired data decoded by
sound.
int sampleIndex=0;
boolean nullReached=false;
while (!nullReached) {
digit[0]=getDecimalDigit(samples[sampleIndex].getValu
e(),0);
digit[1]=getDecimalDigit(samples[sampleIndex+1].getV
alue(),0);
digit[2]=getDecimalDigit(samples[sampleIndex+2].getV
alue(),0);
  int asciiValue=digit[0]+digit[1]*10+digit[2]*100;
  if
(digit[0]==digit[1]&&digit[1]==digit[2]&&digit[2]==0)
  nullReached=true;
  message+=(char) asciiValue;
  sampleIndex+=3;
}
```

```
BASE64Decoder decoder=new BASE64Decoder();
byte[] parameters=decoder.decodeBuffer(message);
// where message contains the encoded parameters.
```

Due to access control TTPA is not privileged to decode the sound file containing the private key as it only belongs to the access privilege of client admin. However it will be decoded in similar fashion as other parameters. Once the decoding process is accomplished, TTPA admin will extract the client admin's public key and VMd and provide it to the server and then server will start the process of calculating the fresh metadata by generating the digital signature for each block and the entire file as follows:

```
sigNew=Signature.getInstance("SHA1withDSA","SUN"
);
sigNew.initVerify(publicKey);
FileInputStream fis=new FileInputStream(fileName);
BufferedInputStream bufin=new
BufferedInputStream(fis);
byte[] buffer=new byte[1024];
int len;
while (bufin.available()!=0){
  len=bufin.read(buffer);
  sigNew.update(buffer,0,len);
};
bufin.close();
```

When new signatures are generated, TTPA admin will request the cloud server to start a comparison process to verify the integrity as follows:

```
boolean
verifies=signofFileorBlockNumber.verify(sigNew);
if(verifies==true){
  report.write("Integrity is Maintained");
}else{
  report.write("Integrity is Violated"+'\n');
}
```

The auditing process will be recorded and a report will be shared among all associated entities to view the auditing results. If TTPA report encounters integrity violation, a request will be sent to cloud admin for successful recovery of violated records from the backup cloud storage.

3.3.5. Data Backup and Recovery Process

We assumed that client's primary storage on cloud is Kuala Lumpur while secondary or backup location is New York datacenter. Under some unwanted circumstances such as natural disasters, malicious attacks or un-wanted modifications, data will be recovered from the New York datacenter without any loss or leakage. During the auditing process if TTPA admin identifies integrity violation, it will not generate success signal for client to move on with further process, unless data is successfully recovered from backup storage to its state of correctness. By viewing the auditing reports cloud admin will initiate the backup and recovery process. If entire file or a block is violated, it will be recovered and there is no need to recover the un-violated records. This process will take place by using following code snippet:

```
byte[] buff=new byte[size];
int bytesRead;
originalFile=new File("File.txt");
originalFileStream=new FileInputStream(originalFile);
backupFile=newFile("BackupFile.txt");
backupFileStream=new FileInputStream(backupFile);
fw=new FileWriter("Recovered.txt");
bw=new BufferedWriter(fw);
String newData;
int i=1;
while
((bytesRead=originalFileStream.read(buff,0,size))>0){
 if(i<5){
   if(i==1){
     backupFileStream.read(buff,0,size);
     newData=new String(buff);
     bw.write(newData);
   }else{
    newData=new String(buff);
    bw.write(newData);
   }
 }
 i++;
} // end while
bw.close();
```

Once the violated, lost or damaged records are recovered back, TTPA admin will re-initiate the auditing process to verify its correctness, since data is successfully backed up, the auditing results will be positive and TTPA will send success message to client for further tasks such as downloading, uploading new files or processing. However the data stored on cloud is homomorphically encrypted, for the decrypted downloading process, client needs to extract his private key from encoded sound file.

3.4. Advantages of the Proposed Technique

Data owner has sufficient degree of control over his data since the client admin is privileged to perform encryption, decryption and metadata generation tasks. Performing these tasks does not require in-depth and technical knowledge of cryptography, security or cloud computing. An intermediate IT admin can handle these operations efficiently. Unlike other cloud systems, client admin is not required to encrypt the data before sending it for cloud storage, using our system, data will be encrypted automatically while it is being uploaded but before it arrives to the cloud storage. Data is encrypted in real-time directly from the stream without being stored at any location. When data arrives at the cloud storage, it will be fully encrypted. Similarly for the decryption process, data is not decrypted at the CSP's end and then transferred to the client because this could raise the concerns of confidentiality violation. Data will depart from cloud storage as encrypted and it will be decrypted in real-time from the download stream. Contents will be downloaded or viewed once arrive at the client's end. With the use of RBAC and random security access generator, client can ensure that only authorized users are accessing the system as per their privilege, this will provide sense of satisfaction to the client.

For performing any transaction, client admin is not required to decrypt the records due to the implementation of partial homomorphism cryptography so it saves time and bandwidth cost. In order to provide client full advantage of acquiring a cloud storage service, using our approach client admin will store the VMd, private and public keys with TTPA, not at their local machine, however these parameters are encoded and can be decrypted only by the privileged user. For the integrity checks, TTPA will initiate an efficient auditing process, where it will be easy to determine the location of violated data from a huge file so as to enable efficient recovery. During the recovery process, cloud admin will check the auditing report to determine the actual data that needs to be recovered instead of recovering the entire file.

Table 1. Evaluation results

Attack ID	Type	Issued by	Attacked assets and operation	Solution to preserve privacy	Final data status
A001	External	External hacker	Data while uploading	Partial homomorphic cryptography, auditing	Intact
A002	Internal external	Cloud admin external hacker	Data at storage	Process, RBAC partial homomorphic cryptography, auditing process, data backup and recovery, RBAC	Intact
A003	Internal external	Cloud admin external hacker TTPA admin	Private key, VMd at storage	Sound steganography, RBAC	Intact
A004	External	External hacker	Private Key, VMd while transferring	Sound steganography, RBAC partial	Intact
A005	External	External hacker	Data while downloading	Homomorphic cryptography, RBAC	Intact

4. EVALUATION

In order to verify the objectives of this research i.e., preserving the privacy of client's data at off-premises cloud storage, we evaluated the security of implemented system using penetration testing strategy in a lab based environment by creating a network of three workstations (one for each user). The system is checked against various attacks that might take place by an external hacker or malicious internal such as cloud or TTPA admin. The evaluation results are represented in **Table 1**. The evaluation process is based on considering the following attack scenarios:

- External hackers launch attacks during the data transmission process to steal the client's confidential records and remotely access the cloud system in order to exploit the access control privileges of the client admin by stealing his authentication credentials
- Internal as well as external hackers breach the physical security barriers deployed by CSP to gain unauthorized and direct access to client's data in order to view, modify or delete the actual contents. They also attack the secure storage of TTPA to steal or manipulate the significant parameters of client i.e., keys and VMd
- A malicious TTPA admin attempts to extract the client's private key to decrypt the data

4.1. Results

During the experiments, we identified that client's privacy always remains intact despite the attacks launched by several malicious users. For-example if an expert hacker is able to attack the data during the transfer (downloading, uploading) or at the storage it doesn't affects the privacy because before data departs from the client it gets and remains encrypted throughout the entire process even when it is stored or processed at cloud storage. When attackers get access, they are not able to get any meaningful information just beside the cipher text and if an attacker violates the integrity at physical cloud storage, it is immediately identified during the auditing process and data is recovered back to its original state from the backup storage.

Similarly when, TTPA admin wants to extract the private key of client, attacker will not be able to decrypt it because it is encrypted as sound. Also if attacker gets the private key, attacker cannot decipher the client's data, since for decryption, system must perform the decrypt process and this task can only be initiated by the client when successfully logs in with required credentials. Un-authorized users cannot perform any operation, even if they break-in security of login menu they need to request for random security code and the code can be only sent to privileged users under the implemented RBAC.

We concluded that using the proposed technique, besides the threatening attacks, client's privacy i.e., data confidentiality and integrity is preserved at off-premises cloud computing storage.

4.2. Discussion

Various researchers across the globe have formulated valuable contributions in forms of models, techniques and algorithms to overcome the security and privacy concerns of adopting the cloud paradigm. In this research, we analyzed the significant contributions of (Ateniese *et al*., 2007; Wang *et al*., 2010; Venkatesh *et al*., 2012; Ranchal *et al*., 2010; Prasadreddy *et al*., 2011). We proposed an improved and enhanced technique by overcoming the limitations of existing work and acquiring their strengths. For-instance we developed the process of efficient third party auditing process as suggested by (Ateniese *et al*., 2007; Wang *et al*., 2010) but we further enhanced their efficiency by implementing partial homomorphic cryptography where users are not only able to store and audit their data files, they can also perform transaction on their encrypted data. Venkatesh *et al*. (2012) proposed a RSA based cryptography technique which facilitates users to only encrypt their data but this technique does not have capability of processing the encrypted data. Similarly, as (Ranchal *et al*., 2010), argued that involving TPA may associate additional risk to confidentiality of client's data, we overcame this issue by implementing a process of sound steganography which secures the significant parameters of client by malicious activities of TTPA hence user's parameters can be stored without any security concerns. Prasadreddy *et al.* (2011) provided the solution for securing the data of client by storing the data and their associated keys with isolated CSPs, however this is a good suggestion, but we believe that involving multiple CSPs may result in enhancing the communication and security complexity. In order to improve this approach we implemented a concept of encoding the cryptographic keys while at storage and during the transfer. Keys and data can be only decrypted by the relevant privileged authorities due to the implementation of RBAC and SCG. In order to overcome data violation issues, we further enhanced the existing work by implementing the process of data backup and recovery, where data of user is efficiently recovered from secondary or backup cloud storage with any loss or damage.

5. CONCLUSION

This paper provides a data privacy preserving solution for organizations dealing with sensitive information to adopt off-premises cloud paradigm. We proposed and presented an improved technique evaluated it by penetration testing. The results showed that client's data is intact and protected from malicious attackers.

For future work, we would solely focus on privacy concerns to maintain the confidentiality and integrity of client's data. We intend to implement the proposed system on Secure Socket Layer (SSL) to enhance the security capabilities during the data transfer and receiving process to protect the system from man-in-the-middle and other session hijacking sort of attacks. Other plans include, enhancing the functionality of proposed technique, implementing a secure computing environment for multiple clients, TTPA and cloud admins working together to protect the data confidentiality and integrity and analyzing and evaluating the performance of system together with its security and privacy capabilities.

6. REFERENCES

Adrian, D., S. Creese and M. Goldsmith, 2012. Insider attacks in cloud computing. Proceedings of 11th International Conference on Trust, Security and Privacy in Computing and Communications, Jun. 25-27, IEEE Xplore Press, England, pp: 857-862. DOI: 10.1109/TrustCom.2012.188

Ateniese, G., R. Burns, R. Curtmola, J. Herring and L. Kissner *et al*., 2007. Provable data possession at untrusted stores. Proceedings of the 14th ACM Conference on Computer and Communications Security, Oct. 29-Nov. 02, ACM Press, USA., pp: 598-609. DOI: 10.1145/1315245.1315318

Catteddu, D. and G. Hogben, 2009. Benefits, Risks and Recommendations for Information Security.

CSA, 2011. Security Guidance for Critical Areas of Focus in Cloud Computing v3.0. USA.

Francisco, R., S. Abreu and M. Correia, 2011. The final frontier: Confidentiality and privacy in the cloud. Computer, 44: 44-50. DOI: 10.1109/MC.2011.223

Hibo, H., J. Xu, C. Ren and B. Choi, 2011. Processing private queries over untrusted data cloud through privacy homomorphism. Proceedings of the 27th International Conference on Data Engineering, Apr. 11-16, IEEE Xplore Press, Germany, pp: 601-61. DOI: 10.1109/ICDE.2011.5767862

Khayyam, H., Q. Ilkin, B. Vusale and B. Mammad, 2012. Cloud Computing for business. Proceedings of the 6th International Conference on Application of Information and Communication Technologies, Oct. 17-19, IEEE Xplore Press, Georgia, pp: 1-4. DOI: 10.1109/ICAICT.2012.6398514

Ling, L., X. Lin, L. Jing and C. Zhang, 2011. Study on the third-party audit in cloud storage service. Proceedings of teg International Conference on Cloud and Service Computing, Dec. 12-14, IEEE Xplore Press, Hong Kong, pp: 220-227. DOI: 10.1109/CSC.2011.6138525

Mervat, B., S. Brohi, S. Chuprat and A. Jamalul-lail, 2012. A Study on significance of adopting cloud computing paradigm in healthcare sector. Proceedings of the 2012 International Conference on Cloud Computing Technologies, Applications and Management, Dec. 8-10, IEEE Xplore Press, UAE, pp: 65-68. DOI: 10.1109/ICCCTAM.2012.6488073

Prasadreddy, P., T. Srinivasa and S. Phani, 2011. A threat free architecture for privacy assurance in cloud computing. Proceedings of the IEEE World Congress on Services, Jul. 4-9, IEEE Xplore Press, USA., pp: 564-568. DOI: 10.1109/SERVICES.2011.11

Ranchal, R., B. Bhargava, L. Ben and L. Lilien, 2010. Protection of identity information in cloud computing without trusted third party. Proceedings of the 29th IEEE Symposium on Reliable Distributed Systems, Oct. 31 Nov. -03, IEEE Xplore Press, Indian, pp: 368-372. DOI: 10.1109/SRDS.2010.57

Rocha, F and M. Correia, 2011. Lucy in the sky without diamonds: Stealing confidential data in the cloud. Proceedings of the 41st International Conference on Dependable Systems and Networks Workshops, Jun. 27-30, IEEE Xplore Press, Hong Kong, pp: 129-134. DOI: 10.1109/DSNW.2011.5958798

Shucheng, Y., C. Wang, K. Ren and W. Lou, 2010. Achieving secure, scalable and fine-grained data access control in cloud computing. Proceedings of the 30th International Conference on Computer Communications, Mar. 14-19, IEEE Xplore Press, USA., pp: 1-9. DOI: 10.1109/INFCOM.2010.5462174

Venkatesh, M., M.R. Sumalatha and C. SelvaKumar, 2012. Improving public auditability, data possession in data storage security for cloud computing. Proceedings of the International Conference on Recent Trends in Information Technology, Apr. 19-21, IEEE Xplore Press, India, pp: 463-467. DOI: 10.1109/ICRTIT.2012.6206835

Wang, C., Q. Wang, K. Ren and W. Lou, 2010. Privacy-preserving public auditing for data storage security in cloud computing. Proceedings of the 30th International Conference on Computer Communications, Mar. 14-19, IEEE Xplore Press, USA., pp: 1-9. DOI: 10.1109/INFCOM.2010.5462173

Yashpalsinh, J. and K. Modi, 2012. Cloud computing concepts, architecture and challenges. Proceedings of the International Conference on Computing, Electronics and Electrical Technologies, Mar. 21-22, IEEE Xplore Press, India, pp: 877-880. DOI: 10.1109/ICCEET.2012.6203873