Research Journal of Applied Sciences, Engineering and Technology 8(20): 2103-2109, 2014
DOI:10.19026/rjaset.8.1204
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 17, 2014 Accepted: July 13, 2014 Published: November 25, 2014

Research Article

An Efficient RSA Based Data Integrity Schema in Secured Cloud Storage

S. Hemalatha and Dr. R. Manickachezian
Department of Computer Science, N.G.M. College, Pollachi, Coimbatore, Tamil Nadu, India

Abstract: This study aim is to address the problem of outsourcing data. Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software and information are provided to computers and other devices as a utility over a network. The imperative feature of cloud services is that the user data are frequently processed remotely in unidentified machines that users do not own or operate. Due to this reason, the data owner loss their original data and it becomes an important obstruction to the extensive implementation of cloud services. This study introduces the term data accountability based encryption and users can substitute to a Third-Party Auditor (TPA) for the verification of data integrity problem. This study proposes the security of the outsourced data in cloud by generating keys for each cloud user attribute with Attribute Based Encryption (ABE) using Efficient RSA (Rivest, Shamir, Adleman) schema and Ring Based Fully Homomorphic Encryption (RBFHE) schema for uploading user data.

Keywords: Cloud computing, data accountability, data integrity, data storage, public auditability

INTRODUCTION

Cloud computing has become one of the most essential technology in today's scenario which facilitates cloud users to share their services over the internet access. Due to the growth and development of this paradigm, privacy and protection has become one of the most active research areas (Pearson, 2009; Hemalatha and Manickachezian, 2013). The privacy protection data has been a greatest concern in the industries as there is lot of threats imposed by various intruders (Pearson *et al*., 2009a). In cloud environment, data can be stored vaguely and the services can be shared among different cloud users in a dynamic manner though maintaining privacy and protection in this dynamic environment is a very challenging task. A number of aspects like data responsibility, privacy and protection together with managing of individual information have to be taken into consideration while designing a cloud environment. So, the cloud user requires sharing data without releasing privacy data to intruders. Thus, primary data protection becomes an essential aspect in the entire cloud computing environment.

In this study, the problem of security issues in cloud computing has been analyzed through data protection and the security issues related to user stored information are solved through specific mechanism. In cloud environment, personal user information facing several data integrity (Cloud Security Alliance, 2010) problems both within and outside environment. It appears from time to time through number of violates for each cloud services (Arrington, 2006; Kincaid, 2008; Amazon.com, 2008). Even though outsourcing statistics to the cloud is cost-effective, it does not instantaneously propose any guarantee on data reliability and accessibility (Hemalatha and Manickachezian, 2012).

Data accountability approaches are essential to protect privacy for each user information in cloud environment. In earlier works (Sundareswaranand Squicciarini, 2012; Crispo and Ruffo, 2001), a Cloud Information Accountability (CIA) structure has been used for data accountability in cloud database. Information accountability mainly focuses on maintaining the data usage transparent and tractable. So, a privacy-preserving third-party auditing procedure is required using data encryption along with data reliability and responsibility.

This study mainly solves the privacy issues in cloud environment. The main goal is to support the data accountability framework. First keys are generated for the user with Efficient RSA and then precede data accountability framework with RBFHE. This study is amongst the few ones to maintain data accountability in an extremely dispersed manner using RBFHE. It focuses on data storage in cloud environment. Besides, with the popularity of cloud computing, several auditing responsibilities from different users might be assigned to TPA. As the individual auditing tasks of each user may be difficult and uncomfortable, a normal requirement is to permit the TPA to proficiently achieve numerous auditing tasks in a group way, i.e., concurrently. Finally, the security result of proposed effort illustration shows

Corresponding Author: S. Hemalatha, Department of Computer Science, N.G.M. College, Pollachi, Coimbatore, Tamil Nadu, India

the fast performance of the design in terms of data integrity and data accountability.

LITERATURE REVIEW

Data accountability structure with automatic storage of cloud data is proposed with logging procedure (Sundareswaran and Squicciarini, 2012). This type of mechanism is highly platform self-determining secured approach and extremely decentralized. It does not need specific data authentication mechanism to secure data, but at this point numerous jar files takes much time to perform the process and more latency is observed by data users.

Pearson *et al*. (2009b) have developed a data accountability approach to deal with security for end users and then extended a privacy approach by integrating privacy manager at server side. The fundamental work lies in incorporating encrypted user data and sending data to cloud. The result of the cloud data is processed by the privacy manager to make public the accurate effects. Though, the privacy manager presents restricted features it does not assure security of the information.

Chun and Bavier (2004) present a layered structural design for handling the trust administration and responsibility in cloud environment. This study mainly controls the trust associations for accountability, security and authentication issues. Moreover, this study focuses on the third-party services to monitor and focus on lower stage of monitoring system resources.

Lee *et al*. (2009) developed an agent based approach in cloud computing environment. Diversified tasks, along with the resource utilization at local technology are followed by fixed software agents. The view of responsibility policies in associated with the security considerations, although it is mostly focused on source utilization and on following sub jobs procedure at various computing nodes than access control.

Wang *et al*. (2010a) presented a distributed auditing protocol with the aim of maintaining the operations of cloud data on dynamic manner to cloud servers, but this approach result in leakage of user data in TPA phase. This approach was modified to support batch auditing technique for multiple owners by Wang *et al*. (2010b). Though, due to the large amount of data, their auditing protocols could justify a serious storage space in the clouds on the server.

In cloud computing, outsourced data are frequently updated by every user through accessing cloud storage data in variety of application principles (Ateniese *et al*., 2008; Wang *et al*., 2012; Erway *et al*., 2009). Therefore, dynamic privacy preserving auditing approach plays a vital role in cloud computing security.

PROPOSED METHODOLOGY

This study presents a cloud data storage service model to perform data accountability and data integrity that involves three major steps as shown in Fig. 1. The

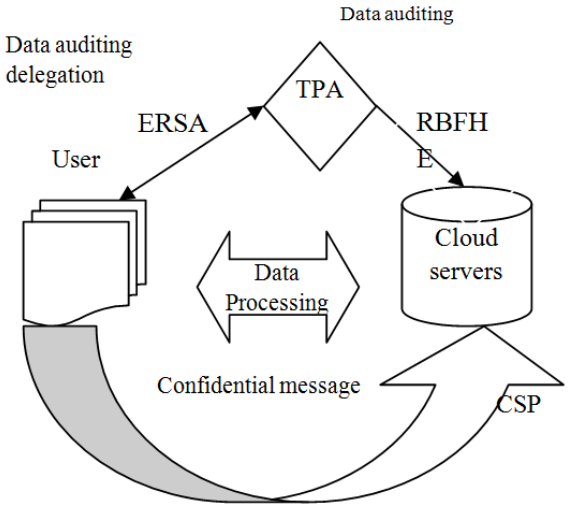

Fig. 1: Cloud data storage

three major entities in the cloud computing service model are cloud user, cloud server and Third Party Auditor (TPA). The cloud user is who interacts with TPA and storage of their data in the cloud. In cloud server, the service of the cloud server is handled by the Cloud Service Provider (CSPs) and has significant storage space and calculation resources. The TPA is responsible to send data to cloud servers and evaluates the cloud storage examination service dependability based on the cloud user request in CSP. Each and every cloud user depends on cloud services from cloud storage database and protection. They might also dynamically cooperate through the cloud server to contact and continuous update their stored data for variety of applications. As the user has no longer storage on local database, it is tough for users to make sure that their information are being properly stored and maintained. In order to accumulate the calculation, resources as well as the online load balancing by the periodic verification of storage space and data integrity are maintained by TPA.

Maintenance of data integrity and data accountability requires a particular encryption approach to receive data without any original information loss for each cloud user. In order to conquer the problems of data integrity and data accountability from cloud storage database as well as cloud environment, data accountability is a technique where the cloud user's data is protected on cloud. In this method, the user's data is in the encrypted form and evaluation is done on encrypted data. The privacy manager formulates the understandable data from outcome of assessment manager to obtain the accurate result.

In obfuscation, data is not present on service provider's mechanism, so the data is protected on cloud. In order to attain data integrity and verification of each user, Efficient RSA encryption is presented in which four keys such as public key, private key, secret key and master key are generated for every user log record. The log record of the user is generated once they login

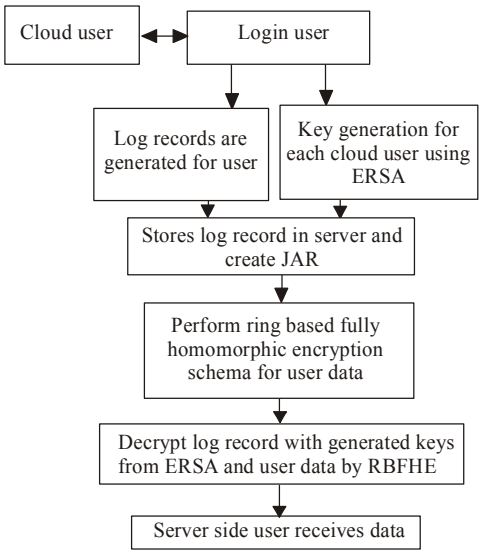

Fig. 2: Architecture of proposed framework

on the server side. User attributes are encrypted based on Attribute Based Encryption (ABE) scheme. Log records of each cloud user's files are generated by using the logging mechanism. In general, log record is obtained in the following form:

$$r_i = (\text{number of attributes}) =< \text{user name}, \text{user id}, \text{location}, \text{date}, \text{time}, \text{year} > \quad (1)$$

The key is generated for each user log record using ERSA. Here each record r_i indicates the number of user and key values are generated from ERSA. The data owner will get a key from Efficient RSA with which he/she is legitimate to upload the data. The uploaded data files of user data will be stored in the equivalent database JAR file along with generated keys. RBFHE is performed for user data after JAR files are generated. The entire representation of the study is shown in Fig. 2.

Efficient RSA encryption for data integrity: Efficient RSA has been used for data intergrity process, that make use of the linear assembly of array r_i, cloud user as n blocks with principles and individual user that was intentionally chosen as arbitrarily from the sphere of numeral value mod n (Aboud *et al*., 2008). The key series of well-organized RSA is significantly important and can essentially be used through Hill cipher technique to acquire additional inflexible scheme. KeyGen is a key generation algorithm that is run by the user to setup the scheme. In this step user defines the public, private and secret key factors for each and every cloud user by implementation of key generation part KeyGen and preprocesses the data file F by using RBFHE to create the authentication metadata process. The differentiation of unique RSA and well-organized RSA is the estimation of key φ ( ) value of cloud user in key generation procedure. In efficient RSA, estimation of keys φ (n) is defined as follows:

$$\varphi(n, a) = (p^a - p^0) \ldots\ldots (p^a - p^{a-1}) + (q^a - q^0) \ldots\ldots (q^a - q^{a-1}) \quad (2)$$

The above mentioned key estimation value improves the security that is, data integrity of each cloud user than general RSA algorithm. In this efficient RSA algorithm, r has been defined as random integer number r for each cloud user. As it was explained previously, it is estimated that the exponent e in well-organized have be the same size as (n). Moreover, exponent d has been calculated according to the Efficient RSA algorithm and the assessment of $\gamma\,(n, h)$. However, in the proposed algorithm the encryption procedure has been performed for cloud user and the key values are created for each user. The decryption process is dissimilar from usual decryption process in RSA because here the keys are created using Improved RSA algorithm. In an ERSA system, user keys are created based on user attributes and a specific key of cloud user can decrypt a specific user data only if there is a match between the keys generated from ERSA .The data files are encrypted using Ring-Based Fully Homomorphic Encryption (RBFHE) of the cipher text and decryption of original file are also performed by using RBFHE schema. The data integrity is verified by using keys that are created from Efficient RSA. So the encryption and decryption will occur like double encryption process.

Key generation algorithm:

1. Randomly choose two prime number for each cloud user, two prime numbers p, q and calculate $n = p.q$
2. Calculate $\varphi(n) = (p-1)(q-1)$
3. Calculate $\varphi\,(n, a) = (p^a - p^0) \ldots\ldots (p^a - pa{-}1{+}qa{-}q0 \ldots\ldots qa{-}qa{-}1$
4. Select, r has been defined as random integer number r for each cloud user such as $1 < r < n$ and $gcd\, r, \varphi$ =1 and $gcd\, r$, $\gamma = 1$ (must be a smallest integer number)
5. Calculate exponent e encryption key for each cloud user such as $r.e \equiv 1\ mod\ \varphi$ ( ) and $1 < e < \varphi\,(n)$
6. After encryption performed then call RBFHE schema for user data
7. Calculate d for each cloud user whose key is generated from encryption part to retrieve and verify data integrity such as $d.\ e \equiv 1\ mod \gamma$ ( ) and $1 < d < \gamma\,(n), 1 < sk < \gamma\,(n),.\ 1 < mk < \gamma\,(n)$

8. Public Key e. Private Key, d, Secret key sk, master key mk
9. Encryption procedure
10. Suppose user A needs to distribute information to cloud user B (represent m as an integer in the range of $0 < M < n$ from log record)
11. User B send a public key, secret key and master key generated from Efficient RSA to user or data owner A, will retrieve data from access point
12. $c = ((m^e mod\ n)^e\ mod\ n)$
13. Decryption procedure
14. User B will decrypt data from data accountability step and then verify user result with key generated from Efficient RSA
    $m = ((c^r mod\ n)^d\ mod\ n)$

After that the completion of Efficient RSA algorithm then perform data accountability for stored user data along with attribute based key results in JAR file.

Third Party Auditor (TPA): If the data owner contains large amount of the outsourced data and the task of examination of data for each data owner, it becomes a difficult task and more costly for data owners. The communication amongst the data owners and cloud servers requires a specific auditor, which can be performed through third party auditing. It provides a cost-effective schema for enabling the trust amongst data owner and cloud server. In order to overcome the limitations of resource calculations of the new users in the cloud environment, TPA has been introduced for better reliability and confidentiality. Every time the data altered has been identified throughout the storage correctness authentication. The third party auditor achieves the each auditing process. The set of policies can be applied to data owner to communicate through cloud servers by using the third party auditing. The data can be downloaded at one moment and then data can be downloaded at an exact time period.

Ring-Based Fully Homomorphic Encryption (RBFHE): In ring based fully homomorphic encryption schema, the generated JAR files of user data are encrypted. This ciphertext comprises of only a single ring element as opposed to the two or more ring elements for schemes based purely on the (ring) learning with errors. The approach is scale-invariant and therefore avoids modulus switching and the size of ciphertexts is one ring element. The cloud user data is cd_n and $\varphi(n, a)$. The most important structure is the ring R. Let d be a positive integer and define $R = Z[r_i] / (\phi_d (cd_n))$ as the ring of polynomials with integer coefficients modulo the d-th cyclotomic polynomial $\phi_d (cd_n) \in Z[cd_n]$. The degree of ϕ_d is $n = \varphi(d)$ where φ is Euler's totient function for data accountability for each user data file cd_n. The elements of R that is user data file cd_n can be uniquely represented by all polynomials in $Z[cd_n]$ of degree less than n. Arithmetic in R is arithmetic modulo $\phi_d (cd_n)$ which is implicit whenever write down terms or equalities involving elements in R. The arbitary coefficient that belongs to the log file in R:

$$a = \sum_{i=0}^{n-1} a_i\, cd_n^{\,i} \in R\ ||a||_\infty = \max_i\{|a_i|\} \qquad (3)$$

where, $a_i \in Z$ and identifies a with its vector of coefficients and choose maximum record based encryption data with $\mathbb{R}^n$ to measure the size of elements in R. When multiplying two elements $g, h \in R$, the norm of their product g, h expands with respect to the individual norms of g and h. The maximal norm expansion that can occur.

$\delta = \sup\left\{1\left|\frac{|g.h||_\infty}{||g||_\infty||h||_\infty}\right.\right\}; g, h \in R$ This is a ring constant.

Let χ be a probability distribution on R that samples small elements $a \leftarrow \chi$ with high probability e.g., a discrete Gaussian distribution The distribution χ on R is called B-bounded for some $B > 0$ if for all $a \leftarrow \chi$ and have $||h||_\infty$<B, i.e., a is B-bounded.

A specific example of a distribution on R is introduced. Initially, the discrete Gaussian distribution $D_{z,\sigma}$ is defined with mean 0 and standard deviation σ over the integers, which assigns a probability proportional to $\exp(-\pi |cd_n|^2/\sigma^2)$ to each data file $cd_n \in Z$ and when d is a power of 2 and $\phi_d (cd_n) = cd_n^{\,n} + 1$ take χ be a spherical discrete Gaussian probability distribution $\chi = D_{Z^n,\sigma}$ in which each coefficient of record is sampled according to the one dimensional distribution. The distribution is used in many fully homomorphic encryption schemes based on RBFHE to sample random error polynomials that have small coefficients with high probability. Such polynomials are a significant part of the noise terms used in the encryption process to deduce meaningful bounds on noise size and noise growth during the homomorphic operations.

Basic public key encryption scheme that is the foundation for the leveled schemes of the next sections. The scheme is parameterized by a modulus q and a plaintext modulus $1 < t < q$. Ciphertexts are elements of $R = Z[cd_n] / (\phi_d (cd_n)$ and plaintexts are elements of /tR. Secret keys and errors are generated from different distributions, for example Gaussian distributions of different width. The secret key is derived from the distribution key χ_{key} and errors are sampled from the distribution χ_{err}.

The basic encryption and decryption steps of the record or user information are define as below.

Basic: params gen (λ): Given the security parameter λ, a positive integer d that determines R, modulo q and t with 1<t<q and distributions χ_{key}, χ_{err} on R output of decrypted data $(d, q, t, \chi_{key}, \chi_{err})$.

Basic: keygen $(\mathbf{d}, \mathbf{q}. \mathbf{t}, \boldsymbol{\chi}_{\mathbf{key}}, \boldsymbol{\chi}_{\mathbf{err}})$: Sample of the cloud data $cds'g \leftarrow \chi_{key}$ and let $cd_n = [tcd_n{}' + 1]_q$ (Here cdn instead of r) if cdn is not invertible to modulo q choose new rs'. Compute inverse $cds^{-1} \in R$ of r modulo q and set $h = [tgcd_n{}^{-1}\varphi(n, h)]_q$.

Output the public and private key pair $(puk, prk, sk, mk) = (h, rs, \varphi(n, a)) \in R^2$.

Basic. encrypt (a, m): The message space is Rt/R for message m+tR+ φ (n, a) choose $[m]_t$ as its representative. Sample s, $e \leftarrow \chi_{err}$ and output the ciphertext of original records:

$$c = [[q/t][m]_t + e + as + \varphi(n, a)]_q \in R \quad (4)$$

The message or information is defined as:

$$m = [[t/q \,.\, [cds]_q]]_t \in R \quad (5)$$

Often refer to a message as an element m in the ring R although the message space is R/tR, keeping in mind that encryption always takes place on the representative $[m]_t$ and that by decrypting, all that can be recovered is m modulo t. The following lemma states conditions for a ciphertext c such that the decryption algorithm outputs the message m that was originally encrypted data.

Let q, t and $\Delta = \left[\frac{q}{t}\right]$ be as above and let $cd, r, m \in R$. If there exists $v \in R$.

Such that $cds_c = \Delta\,[m]_t + v \pmod q$ and $||v||_\infty < (\Delta - cd_t(q))/2$ then basic. Decrpt (cd, c) = $[m]_t$ under the secret key cds.

Of course, for any given c, cds and m, there always exists a $v \in R$ such that $cds_c = \Delta\,[m]_t + v \pmod q$. But only a v of small norm allows one to recover $[m]_t$ from c. Since they always free to vary v modulo q, i.e., to add any multiple of q to it, then choose v to be the canonical element $[v]_q$. This means the equation v with the smallest possible norm among all polynomials that satisfy the equation. The following lemma derives a bound on the inherent noise in a freshly encrypted ciphertext output by Basic Encrypt, assuming bounds B key on the key and Berr on the error distributions.

Note that since$cds', g \leftarrow \chi_{key}$, $||rs'||_\infty, ||g||_\infty \leftarrow tB_{key}$ and $||cds||_\infty = ||1 + tcd_n s'||_\infty < tB_{key}$ since $t \geq 2$. Let the key and error distributions are B_{key}-bounded and B_{err}bounded, respectively.

Given $m \in R$, a public key $a = [tgrs^{-1}\varphi(n, a)]_q$ with secret key $cds = [1 + tcds']_q, cds', g \leftarrow \chi_{key}$ and let $c = Basic.Encrypt(a, m)$. There exists $v \in R$ such that $rsc = \Delta\,[m]_t + v \pmod{q(\varphi(n, a))}$ and $||v||_\infty, < \delta t B_{key}(2B_{err} + 1cds_t(q)/2)$. Then now integrate the fully homomorphic encryption scheme operations and deduce bounds on the noise growth that occurs during these operations.

RBFHE: parametergen (λ): Given the security parameter λ output of the parameter with log file encrypted using $(d, q, t, \chi_{key}, \chi_{err}, w)$ Where $(d, q, t, \chi_{key}, \chi_{err}) \leftarrow$ BasicParamgen (λ) and w>1 is integer,

RBFHE. keygen $(d, q, t, \chi_{key}, \chi_{err}, w)$

computeh, rs $\leftarrow$ Basic. Keygen $(d, q, t, \chi_{key}, \chi_{err})$

ample $e, s \leftarrow \chi_{err}^{\ell^3_{w,q}}$ compute $\gamma = [cds^{-1}P_{w,q}\left(D_{w,q}(cds) \otimes D_{w,q}(cds)\right) + e + h.s]_q \in R^{\ell^3_{w,q}}$

RBFHE. encrypt (puk, prk, sk, mk, evk) $= (a, cds, \gamma)$

RBFHE. encrypt (puk, prk , mk, sk, m) to encrypt $m \in R$

RBFHE. Decrypt (sk, pk, mk, c) to output the message encrypt $m \leftarrow$ Basic. Decrypt (sk, mk, prk, c) $\in R$

RBFHE. KeySwitch $(\tilde{c}_{multi}), evk)$: output $[\langle D_{w,q}(\tilde{c}_{multi}), evk\rangle]_q \in R$.

RBFHE. add(C_1, C_2): Compute the addition of C_1, C_2 as $C_{add} = [C_1 + C_2]_q$

RBFHE. multi (C_1, C_2, evk) Compute $\tilde{c}_{multi} = \left[\left[\frac{t}{q}P_{w,q}(C_1) \otimes P_{w,q}(C_2)\right]\right]_q \in R^{t^2_{w,q}}$ and output $c_{multi} =$ RBFHE. multi $(\tilde{c}_{multi}, evk)$

Given two ciphertexts $C_1, C_2 \in R$ which encrypt two messages m_1, m_2 with inherent noise terms v_1, v_2 their sum modulo q, $C_{add} = [C_1 + C_2]_q$ encrypts the sum of the message modulo t $[m_1 + m_2]_t$ and rewrite this as $[m_1 + m_2]_t + tcd_{add} = [m_1]_t + [m_2]_t$ for some $cd_{add} \in R$ with $||cd_{add}||_\infty \leq 1$:

$$\begin{aligned} &Cds[C_1 + C_2]_q = cdsC_1 + cdsC_2 = \Delta([m_1]_t + [m_2]_t] + (v_1 + v_2) \\ &= \Delta([m_1 + m_2]_t + tcds_{add} + (v_1 + v_2)) \pmod q \end{aligned} \quad (6)$$

This means that the size of the inherent noise v_{add} of c_{add} is bounded by:

$$||v_{add}||_\infty \leq ||v_1||_\infty + ||v_2||_\infty + cds_t(q) \quad (7)$$

Homomorphic Multiplication operation is divided into two parts. The first part describes a basic procedure to obtain an intermediate ciphertext that encrypts the product $[m_1\ m_2]_t$ modulo t of two messages m_1 and m_2. However, the intermediate ciphertext cannot be decrypted with Basic: Decrypt using the secret key rs. The second part performs a procedure which allows a public transformation of this intermediate ciphertext to a ciphertext that can be decrypted with rs. This latter procedure was introduced in Pearson *et al*. (2009b) in the form of relinearization and was later expanded in Pearson *et al*. (2009a) into a method called key switching, which transforms a ciphertext decryptable under one secret key to one decryptable under any other secret key. For our analysis, assume that χ_{key}, χ_{err} are

B_{key}-bounded and B_{err} bounded, respectively. Even if work with unbounded Gaussian distributions, this is a valid assumption since elements drawn from either distribution have bounded norm for suitable bounds with high probability. $\text{RBFHE.multi}(C_1, C_2, evk)$ compute:

$$\tilde{C}_{multi} = \left[\left[\frac{t}{q} P_{w,q}(C_1) \otimes P_{w,q}(C_2)\right]\right]_q \in R^{t^2_{w,q}} \quad (8)$$

The second part in the homomorphic multiplication procedure is a key switching step, which transforms the ciphertext $\tilde{c}_{multi}$ into a ciphertext cmult that is decryptable under the original secret key, private key and master key cds:

$$evk = [cds^{-1} P_{w,q}\left(D_{w,q}(cds) \otimes D_{w,q}(cds)\right) + e + a.s]_q \quad (9)$$

Output by RBFHE.Keygen where $e, s \leftarrow \chi_{err}^{-w,q}$ are vectors of polynomials sampled from the error distribution χ_{err} and $[.]_q$ is applied to each coefficient of the vector. Note that this key is a vector of quasi-encryptions of $cds^{-1} P_{w,q}\left(D_{w,q}(cds) \otimes D_{w,q}(cds)\right)$ that depend on the secret keyrs, under its corresponding public key and that it is made public because it is needed for the homomorphic multiplication operation.

Finally the decrypted information are verified by using the Efficient RSA based schema and the logs are periodically pushed to the data owner (or auditor) by the harmonize:

- It ensures that the size of the log files does not explode
- It enables timely detection and correction of any loss or damage to the log files

Concerning the latter function, notice that the auditor, upon receiving the log file, will verify its cryptographic guarantees, by checking the records' integrity and authenticity. By construction of the records, the auditor will be able to quickly detect forgery of entries, using the checksum added to each and every record.

EXPERIMENTAL RESULTS

The experimental result evaluates the performance of the proposed approach. The experimental evaluation has been carried out through CloudSim (Calheiros *et al*., 2009, 2011) framework that permits to model and perform experimentation of the cloud computing infrastructure in various applications (Calheiros *et al*., 2011).

In this section, experimentation result of privacy preserving auditing results of MAC and ERSA-RBFHE Approaches are examined based on the simulation results done using CloudSim. The performances of the existing MAC and proposed ERSA-RBFHE encryption methods are examined through the parameters like execution time and throughput.

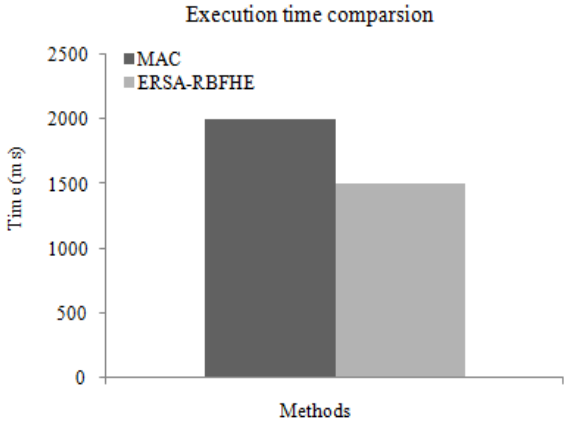

Fig. 3: Comparison of execution time

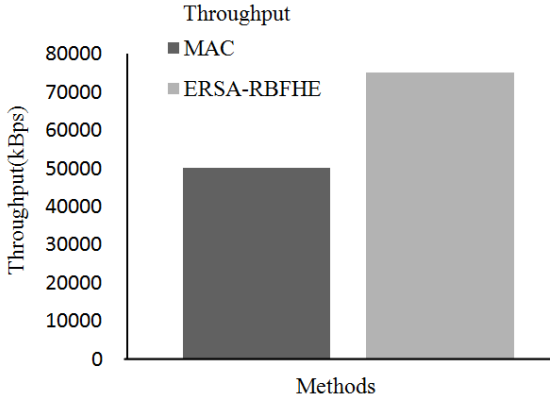

Fig. 4: Throughput comparison

Figure 3 measures execution time comparison of RSA and RBFHE schema which results in better security than MAC. According to the experimental results, the execution time of MAC is high when compared to propose ERSA-RBFHE, due to best security results. From the result it is observed that the proposed method gives less execution time.

Throughput represents the data transfer rate over the network successfully. The result of throughput is the high number of data transferred from cloud user to server. It improves security level of system. Figure 4 shows the result of MAC, ERSA-RBFHE throughput. It shows that proposed ERSA-RBFHE have higher throughput result than MAC. It shows that proposed system has highest data transfer than existing methods.

CONCLUSION

In this study, an efficient privacy preserving auditing mechanism has been proposed which combines both ERSA and RBFHE schema. The security level of proposed schema provides higher security considerations than MAC method. Efficient RSA is

introduced in this approach to exploit the data integrity throughout the well-organized auditing process. In ERSA schema, four keys are generated to each log files and generated keys are converted into JAR files to upload their data in the cloud server. Based on role based action control, the user uploads their data in server. Then cloud user data in JAR file are encrypted using RBFHE to increase the security level of both user and their data. It also reduces the leakage of outsourced data for cloud data owner. Experimental result shows that proposed ERSA-RBFHE schema have high throughput rate when compared to MAC schema.

REFERENCES

Aboud, S.J., M.A. Alfayoumi, M. Alfayoumi and H. Jabbar, 2008. An efficient RSA public key encryption scheme. Proceeding of the 5th International Conference on Information Technology: New Generations (ITNG), Las Vegas, pp: 127-130.

Amazon.com, 2008. Amazon s3.Retrieved from: http:// status. aws. amazon. com/ s3-20080720. html (Accessed on: July 20, 2008).

Arrington, M., 2006. Gmail Disaster: Reports of Mass Email Deletions. Retrieved from: http:// www. techcrunch. com/ 2006/ 12/28/ gmail-disasterreports of-mass-email-deletions/.

Ateniese, G., R.D. Pietro, L.V. Mancini and G. Tsudik, 2008. Scalable and efficient provable data possession. Proceeding of the International Conference on Security and Privacy in Communication Networks (SecureComm'08), pp: 1-10.

Calheiros, R.N., R. Ranjan, C.A.F.D. Rose and R. Buyya, 2009. CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services. Technical Report, GRIDS-TR-2009-1, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia.

Calheiros, R.N., R. Ranjan, A. Beloglazov, C.A.F. De Rose and R. Buyya, 2011. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software Pract. Exper., 2011: 23-50.

Chun, B. and A.C. Bavier, 2004. Decentralized trust management and accountability in federated system. Proceeding of the Annial Hawaii International Conference on System Science (HICSS).

Cloud Security Alliance, 2010. Top Threats to Cloud Computing. Retrieved from: http://www. Cloud securityalliance.org.

Crispo, B. and G. Ruffo, 2001. Reasoning about accountability within delegation. Proceeding of the 3rd International Conference on Information and Communication Security (ICICS), pp: 251-260.

Erway, C., A. Kupcu, C. Papamanthou and R. Tamassia, 2009. Dynamic provable data possession. Proceeding of the ACM Conference on Computer and Communication Security (CCS'09), pp: 213-222.

Hemalatha, S. and R. Manickachezian, 2012. Present and future of cloud computing: A collaborated survey report. Int. J. Innov. Technol. Exploring Eng., 1(2).

Hemalatha, S. and R. Manickachezian, 2013. Implicit security architecture framework in cloud computing based on data partitioning and security key distribution. Int. J. Emerg. Technol. Comput. Appl. Sci., 2013: 76-81.

Kincaid, J., 2008. MediaMax/TheLinkup Closes Its Doors. Retrieved from: http://www.techcrunch. com/2008/07/10/media maxthelinkup-closesits-doors/.

Lee, W., A. Cinzia Squicciarini and E. Bertino, 2009. The design and evaluation of accountable grid computing system. Proceeding of the 29th IEEE International Conference on Distributed Computing Systems (ICDCS'09), pp: 145-154.

Pearson, S., 2009. Taking account of privacy when designing cloud computing services. Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD '09), pp: 44-52.

Pearson, S., Y. Shen and M. Mowbray, 2009a. A privacy manager for cloud computing. Proceedings of the International Conference on Cloud Computing (cloudcom), pp: 90-106.

Pearson, S., A.C. Squicciarini and D. Lin, 2009b. Accountability as a way forward for privacy protection in the cloud. Proceeding of the 1st International Conference on Cloud Computing.

Sundareswaran, S. and A.C. Squicciarini, 2012. Ensuring distributed accountability for data sharing in the cloud. IEEE T. Depend. Secure, 9(4): 556-568.

Wang, C., Q. Wang, K. Ren and W. Lou, 2010a. Privacy-preserving public auditing for data storage security in cloud computing. Proceeding of the IEEEINFOCOM, San Diego, CA, pp: 525-533.

Wang, Q., C. Wang, K. Ren, W. Lou and J. Li, 2010b. Enabling public auditability and data dynamics for storages security in cloud computing. Proceeding of the IEEEINFOCOM, pp: 525-533.

Wang, C., Q. Wang, K. Ren and W. Lou, 2012. Towards secure and dependable storage services in cloud computing. IEEE T. Serv. Comput., 5(2): 220-232.