I J C T A, 9(2) 2016, pp. 1169-1178

Public Data Integrity Auditing Using Scram Based Data Sharing In The Dynamic Cloud Environment

B. Gunalan[*] and T. Christopher[**]

ABSTRACT

Cloud storage service is one of the developing technologies which isused to provide the variety of services to different users. In the cloud storage services various users store their information continuously and updating the information for improving the quality of the cloud storage services. During the cloud storage services, the data integrity, confidentiality, revocation of the shared data is one of the main challenges because various users modify the information in the cloud environment. Different cloud storage auditing techniques are used to managing the modified data, auditing cost in the cloud, but it limits these techniques to client read-only data. So, the paper proposes a secret key sharing and a secret key authentication mechanism is used to manage the cloud storage data which is done with the help of the Homomorphic secret sharing and Salted Challenge Response Authentication Mechanism (SCRAM) techniques. These techniques authenticate the user and server secret keys while modifying the cloud file information that increases the error detection probability, efficient user revocation and public auditing. Then the performance of the public auditing and data integrity is analyzed with the help of the experimental results.

Keywords: Cloud Auditing, Cloud storage services, Secret key sharing, Homomorphic secret sharing and Salted Challenge Response Authentication Mechanism, error detection probability, efficient user revocation and public auditing.

1. INTRODUCTION

Cloud data storage use virtualized pools where applications and data are stored in electronic formatewhich are basically hosted by third party. Some of the advantages of data storage in cloud are reduced web application management through automated updates, reduce IT management of data and hardware, provide more flexibility, reduce cloud storage cost and transimmions cost and provide high capacity of cloud storage [1]. In spite of these kinds of advantages, “cloud” lack in some kinds of problems for example privacy, unauthorized data access, data loss, data integrity and so on [2]. Some of the top examples for cloud storage are iCloud by Apple,Google App Engine Blobstore,Google Cloud Storage,EMC Atmos,Windows Azure Storage andAmazon S3.

Data Integrity is a significant process among the other issues of cloud storage. Because the data integrity make sure that the data is of accessible, consistent, correct, and high quality. After transforming the data to the cloud, the data owner belive that their application and data are in secured manner [3]. However, that belive may fail in some points which means the data owner may be delected or altered that data. In other word the group member may altered or delected that data. In this scenario, it is significant to verify if twice data has been delected or tampered.

To validate the data, often the data user must download that data. If the stored data is huge files which means such downloading take much time and it direct to data integrity [4] [5]. The data integrity may

[*] Research Scholar, Government ArtsCollege,Udumalpet, Tamil Nadu.
[**] Assistant Professor, Government Arts College, Udumalpet, Tamil Nadu
E-mail: gunalanb61@gmail.com

become unaffordable in term of increased time and cost of bandwidth, pariticually if need frequent data checks. Thus, in this paperproposes a secret key sharing and a secret key authentication mechanism is used to manage the data storage in cloud which is done with the help of the Homomorphic secret sharing and Salted Challenge Response Authentication Mechanism (SCRAM) techniques. These techniques authenticate the user and server secret keys while modifying the cloud file information that increases the error detection probability, efficient user revocation and public auditing. Additionally, this work mainly focuses a design issue is as dynamic cloud data operation during the data storage in term of data integrity, confidentiality, revocation of the shared data and auditing cost. So here focus on the data dynamics and public auditability.

Todone this data dynamics and public auditability has to build the solution that meet essential requirement link retrievability of cloud data, high efficiency and public data integrity. Due to, contribution of verifier which means Third Party Auditor (TPA) verify two kind of auditability process such as public auditability and private auditability. Although, the public auditability permit anyone can access the data not only for data owner and private auditability achieve greater scheme efficiency. In public auditability scheme has trouble in while keeping private information during the data storage. So the data owner give all privilege to TPA without any commitment because the cloud may not able to frequently check the cloud data integrity. Thus, there is required to create verification protocol which has public auditability.

2. RELATED WORK

In [6] author present the study about cloud data intergrity and security in cloud environment. In this work consider the TPA used to verify the cloud data intergrity during the process of data storage. The method of bilinear aggregate signature is utilized to accomplish the batch auditing. The batch auditing minimize the computation overhead. The new scheme further supports efficient and secure dynamic operation on data blocks which also comprisingdata append, delete and update. Additionaly, in this work the hyper graph model is used to cluster the high dimensional data, and the hyper edges are linked by using the association rules algorithm where frequent item sets found. This work achieves promptness and accuracy, additionally, the server labling and user preference is also included.

In [7] author present an auditing model based on Merkle Hash Tree. This process mainly focused on audition mechanisms which is service over the public and hybrid clouds. This process also used to verifythe data stored integrity in the public clouds.Merkle Hash Tree is built by using Lamport's one time signatures. This signature needs a setup stage, at this stage have to select the secret values and employing a high function such as SHA1to each and every publication.

In [8] author present the public auditability for cloud storage which is done by using TPA. This work mainly proposes a secure cloud storage system with privacy-preserving public auditing process. And also it extend the result, audits are done on multiple cloud users efficiently and simultaneously. The performance analysis and extensive security present the proposed work are promising in term of high efficiency and provably secure. The experimental done on Amazon EC2 instance.

In [9] author deals with multi cloud environments to resolve the cloud security issues and data integrity issues. It mainly process on avoiding the single cloud provider issues such as the single cloud provider has less security and risk of server failure, availability and additionally it has malicious user possibility. These issues are avoided by using multi cloud environment. The multi cloud storage integrity check by using Trusted Third Party (TTP) wih the help of cryptographic algorithm.

In [10] author analysis the cloud storage platform's integrity vulnerability and also presents the repudiation problem in cloud storage system. In this work presents the novel Nonrepudiation (NR) protocol particularly designed in term of cloud computing environement. Here, also dicussednR protocol's robustness against the specific attacks in the network environments. This proposed work concerning the data security in

cloud, before addressing this novel pay-as-your go business model in term of both potential malicious users and unreliable/insecure service providers.

In [11] author present a cooperative Provable Data Possession (CPDP) approach based on homomorphic verifiable response and hash index hierarchy for distributed cloud stoage system which process support data migration and scalability of cloud service. This process consider the multiple cloud service providers to maintain the clients' data. This process also provide multi prover zero knowledge proof system for data security which can satisfy the zero knowledge properties, knowledge soundness and satisfy completeness. Additionally, present the efficient approach for electing the optimal parameter values to reduce computation cost of storage service providers. The experimental results show promising results in term of communication overheards and lower computation when compared with cooperative methods.

3. SYSTEM MODEL

In this propose work consider a cloud system included three different entities such as Thrid Party Auditor (TPA), group users, and cloud server as shown in figure 1. the cloud server is known as third party that gives cloud data storage services to the user's groups. The group of user compriseof a mater user and a number of general users, who is the owner of the shared data and manage the group user's membership function. All group user can modify and access the data and the TPA denotes to any knids of party that checks the data integrity during the cloud storage process.

As this proposed works allows the public integrity audition process with the help of TPA, actually, the TPA can be any cloud user as lengthy as she/he has use the public key. During the auditing process, Once the TPA detects a data corruption she/he will report about this error to other group members. Assume that the data are stored in the form of files than that files are separated into a number of different blocks. For the process of integrity auditing, each and every block is attached with an another new tag such as authentication tag which is originally created by the master user or data owner. When a user modifies or add a block, the user (she/he) updates the appropriate authencication tag wih her/ his own secret key without asking the data owner or mater user.

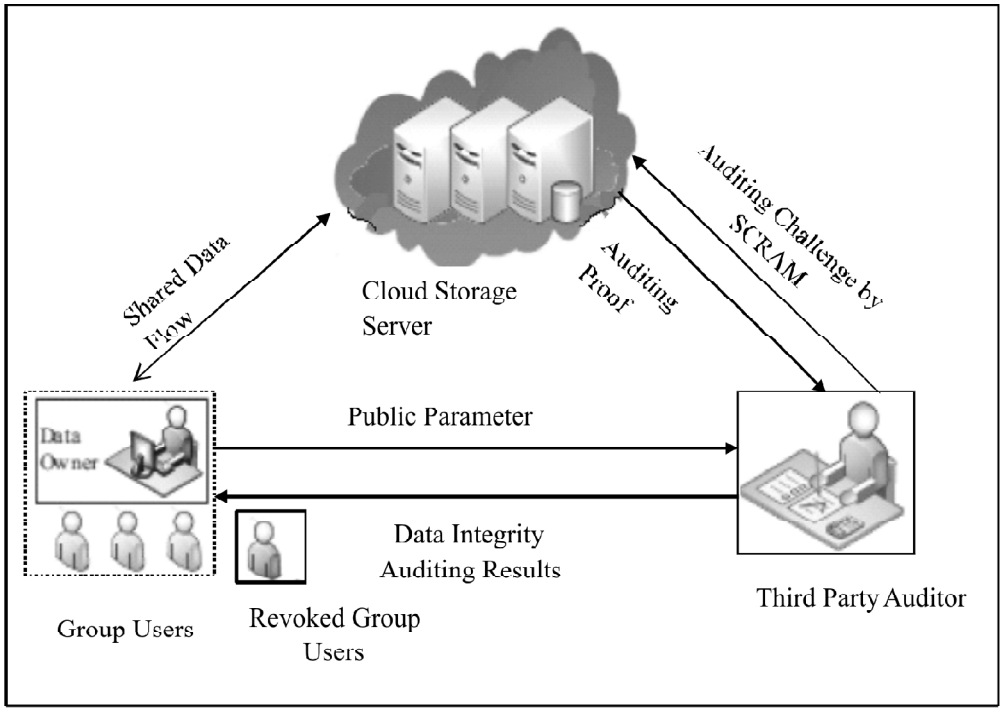

Figure 1: System Model of Proposed Work

Data Owner: who create the cloud data is term as data owner and he/ she have a large amount of data to be stored in the cloud.

Cloud Storage Server (CSS): the cloud storage server is managed by Cloud Service Provider (CSP) to givethe data Storage Service in the cloud. The CSS is separated into two different components such asData Server, Stores the clients data and Management Server, manages the server.

Third Party Auditor (TPA): TPA has capabilities to monitor or manage outsourced data over the entrustment of data owner. The file's hash values are stored at TPA.

Cloud Service Provider (CSP): The CSP has significant computation recourseand storage apace to maintain clients or user data.

So, the paper proposes a secret key sharing and a secret key authentication mechanism is used to manage the cloud storage data which is done with the help of the Homomorphic secret sharing and Salted Challenge Response Authentication Mechanism (SCRAM) techniques.

3.1. HomomorphicSecret Sharing (HSS)

Homomorphic secret sharing system are utilized to perform a process on encrypted data without knowing the private key, the data owner is the only secret key's owner. When the decrypt operation is done, it is similar as the encryption process. In this work, the well-know Homomorphic Secret Sharing system such as Shamir secret sharing is used. But, Shamir secret sharing is only an additvelyhomomorphic but which is not mutiplicativelyhomomorphic. Thus, it have to perform multiplication utilizing Shamir secret shares, a particularly "degree reduction step" is needed.

Sharmir's Secret Sharing (SSS) [12] [13]scheme is formeted to share a *single secret valuess*$_j$ amongst *n server* such that the shares must be extracted from any *k server* so as to *reconstructs*$_j$. The security rests of this schemes on the fact that as a minimum *k points* are required to uniquely reconstruct a degree of polynomial *k*–1. Typically, the coefficents and points utilized in scheme of sharmir can be obtain from the field *F*. However, to utilize this scheme on finite-accuracy machines, here need *F* to the finite field F_p where *p* is defined as the prime.

To share s_j, –1 coefficients $a_1, j, \ldots, a_{k-1}, j$ nominated randomly from F_p.select a prime $p > s_j$, and *k* then construct the polynomial function is as follows:

$$q_j(x) = s_j + \sum_{i=1}^{k} a_h, j^{x^h} mod\ p \tag{1}$$

After this process create a vector $X = (x_1,\ldots, x_n)$ of distinct elements in F_p, and for each and every data server DS_i, find the share $y_{i,j} = q_j(x_i)$. Together, x_i and $y_{i,j}$ form a point $(x_i, y_{i,j})$ over which polynomial $q_j(x)$ passes.

Certain any *k* such points $(x_1, y_1, j), \ldots, (x_k, y_k, j)$, here, can reconstruct the polynomial $q_j(x)$ utilizing Lagrange interpolation:

$$q_j(x) = \sum_{i=1}^{k} y_{i,j} l_i(x)\ \ mod\ p \tag{2}$$

where $l_i(x)$ is defined as the Lagrange basis polynomial is as follows

$$l_i(x) = \prod_{1 \le j \le k,\ \ j \ne i} (x - x_j)(x_i - x_j)^1\ \ mod\ p \tag{3}$$

and $(x_i - x_j)^1$ is defined as the multiplicative inverse of $(x_i - x_j)$ *modulo p*. The secret s_j is the polynomial q_j weighedat $x = 0$, so here get

$$s_j = \sum_{i=1}^{k} y_{i,j} l_i(0) \ mod \ p \tag{4}$$

Given only $k' < k$ shares, and thus only k' points, here cannot read anything about s, then for any value of s, could build a polynomial of degree $k - 1$ that passes over all k' points. Accordingly Shamir's scheme provides perfect, theoretic information security against recuperating s_j from fewer than k shares.

Thus, in order to execute SSS, have to use following operations

WRITE (*X*, *Y*): Write the data *Y* in address *X*.

READ (*X*): Read the data at address *X*.

JUMP (*C*): Transfers control to index *C*, process the branching operation.

LOAD (*H*): Load the instruction in address *H* to the processor.

Let assume both the data items and addresses are secret shared utilizing the same degree polynomials operation', plus degree reduction step is as follows

The SSS- Plus Degree Reduction
Step 1: $procedure\ SSS - \ Degree\ Reduction\ (A, B, C)$
Step 2: $Decrease(A\|\|B\|\|C\|\|P\ C\ +\ 1, 3`, `)$
Step 3: $R1\ \leftarrow\ READ(A)$
Step4: $R2\ \leftarrow\ READ(B)$
Step5: $R\|\|Num\ =\ SSS - SUB(R1, R2)$
Step6: $DECREASE(R\|\|Num, *, `)$
Step7: $WRITE(B, R)$
Step8: $JUMP(Num\ \cdot\ C\ +\ (1\ -\ Num)\ \cdot\ (P\ C\ +\ 1))$
Step9: end

The threedifferent parameters *A*, *B*, *C* where defined as contents at *address B* are subtracted from the substances at *address A*, and the product is stored at *address B*, and then, if the result is *not greater* than 0. PC is defined as the *program counter*.

Although the implementation stores the secret shares hashedscuh as file's hash values, knowing the hashed secret sharesitself is sufficient information to fruitfullyauthenticate against a cloud server. But Anotherissue is that hashing function is not enough for providing a securely storing passwords. Therefore a hash fuction should be salted to avoid against pre-computed hash table look-ups password. Thus, in work the Salted Challenge Response Authentication Mechanism (SCRAM) techniques is used.

3.3. Salted Challenge Response Authentication Mechanism (SCRAM)

Salted Challenge Response Authentication Mechanism (SCRAM) which is part ofSimple Authentication and Security Layer (SASL) protocols.SCRAM is extensivelyutilized for services like Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP) or Extensible Messaging and Presence Protocol (XMPP) and so on.

The proposed SCRAMprotocol as show in figure 2 offers the following functionalities:

All kinds of messages in SCRAM are in text based messages and it comprises value/ attribute pairs divided by commas and each and every attribute has a one-letter name. Addtionally, the process of authernication is stated by a data owner.

Authentication occurs on both sides which means the group members and data owner also checks that the server actually did hold the appropriate password.

The file's hash values is transparent and it can be modified, what creats the protocol forthcoming proof.

The group user's passwords can be hashed and also salted on the server side.

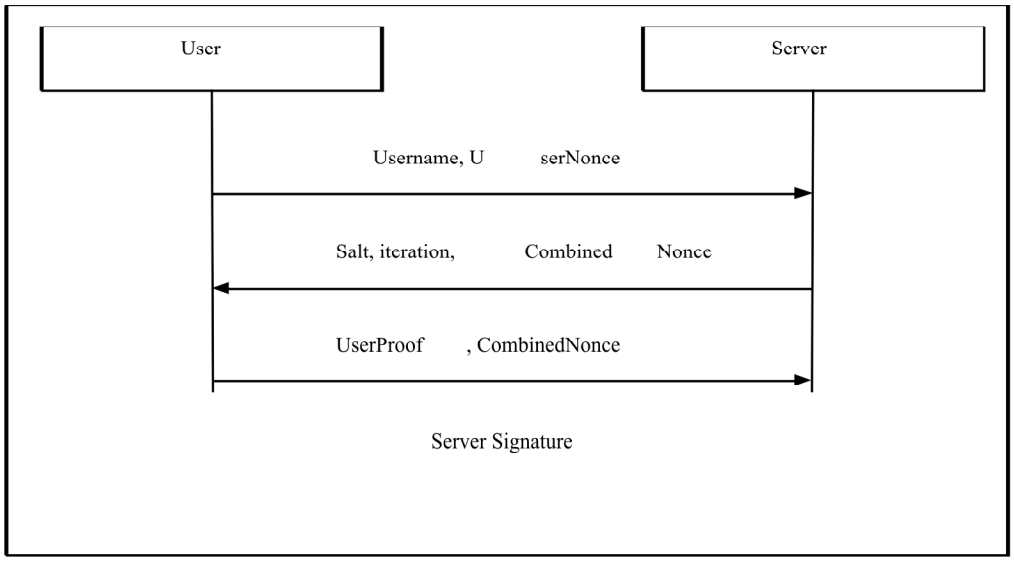

Figure 2: SCRAM protocol's Authentication Session

SCRAM protocol's Authentication process is as follows

Step 1: The user initiates a SCRAM authentication session

The user sends an authentication request to the cloud server which comprisinga random number (named the ClientNonce) and a username utilized to avoid replay attacks.

Step 2: The server issues a challenge

The server initiallyrecovers the credential of user, and replies with a message comprising a CombinedNonceand a salt, iteration count, and a concatenation of the UserNonce and an addedServerNonceproduced by the server.

Step 3: The client replies with a proof

The reply to the challenge is a message comprising the CombinedNonceand UserProof. The UserProofpermits the client to show that it has ownership of the ClientKey.

To find the UserProof, initallyuserfinds the UserKeyand StoredKeyin the similar way as the server when it initially creates the credential of user.

The user then find the UserSignature utilizingthe AuthMessageand the StoredKey. AuthMessage is defind as it just a the server's challenge, user's initial message, and the User's reply (without any UserProof).

$$UserSignature = HSS\ (StoredKey, AuthMessage) \quad (5)$$

The user then compute the UserProof by processing a *bitwise XOR* operation of the UserKey and the UserSignature.

$$UserProof = UserKey \oplus UserSignature \quad (6)$$

As the cloud server can calculate the UserSignatureutilzing the information stored in its HSS, the with the UserKey, which is not stored by the cloud server, avoids the server from being able to fake a valid UserProof.

Step 4: The server verifies the User's proof, and provides own issues

Now the server formate the UserSignature utilizing the StoredKey from the credential of users.Thenthe server can verify the UserProofutilzing the following formula:

$$H(USerSignature \oplus UserProof) = StoredKey \quad (7)$$

If the above equation shows, the server has vaildproof that the user can access to the UserKey. After this process verifying the UserProof, the server creates its own validation proof.

$$ServerSignature = HSS(ServerKey, AuthMessage) \quad (8)$$

Then the server responses to the user with the ServerSignature.

Step 5: The user verifies the proof of server

The user verifies the proof of the server is done by using ServerSignature and ServerKey, then comparing its ServerSignature to the already obtained from the same server. If the Server signature is match, the user has proof that the processing server can access the ServerKey.

4. RESULTS AND DISCUSSION

The proposed cloud secure sharing evaluation process is done by using JAVA with JAVA Pairing-Based Cryptography Library (jPBC) [14] on Amarzon EC2 cloud. On the cloud server, employ nodes running Linux with 32GB memory and 8-core CPU. The group users machines are used as laptops which is using running Linux with 8GB memory, 2.50GHz Intel i5-2520M CPU. The verifier is a set as desktop with running Linux with ASUS Android Iconia A200 Tablet using1-GHz NvidiaTegra2 Dual Core Mobile CPU or 16GBmemeroy and 3.4GHz Intel i7-3770 CPU. In order to compare the proposed work with [15] [16] with the same setup of experiment work.In this part mainly analysis the communiciaton and computation performance of proposed work. All the experimental results signifies the mean of 100 trails.

4.1. System Setup

Initially evaluate the generation of public key and signature for the proposed system. From the result in Figure 3 (a) shows that the public key generation time is comparative to the group size and (b)shows that the signature generation time is comparative to the block size. The group size, since the master user required to create secret keys for each and every group user individually.

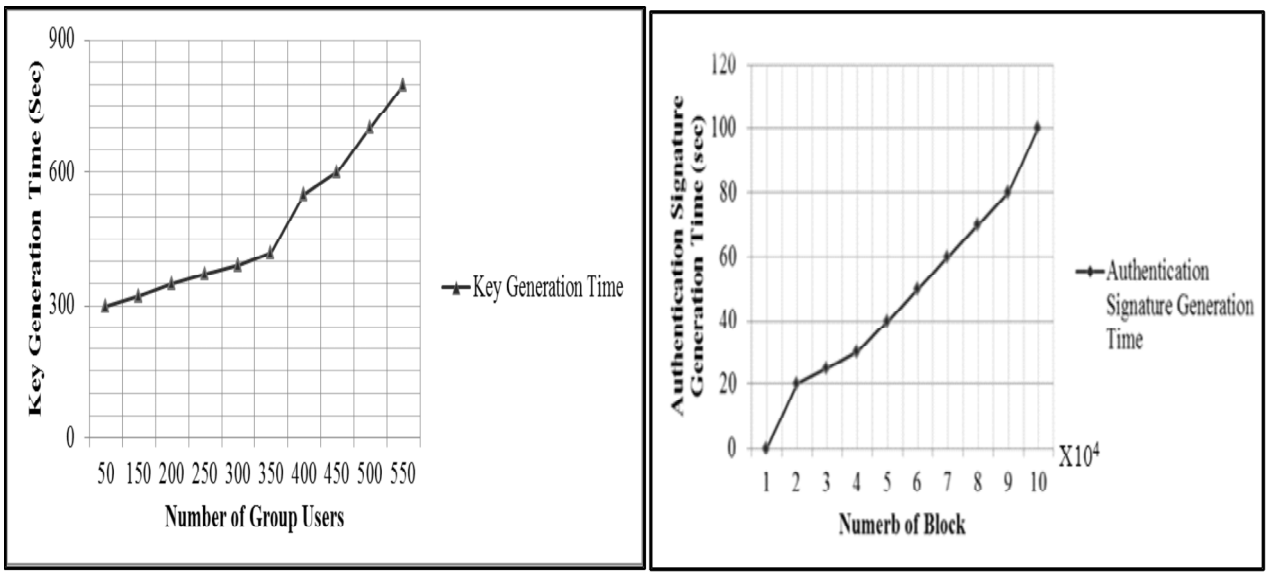

Figure 3: (a) Key Generation Time. (b) Authentication Signature Generation Time

To present the performance of authentication signature generation, different number of blocks in the files from 1000 to 100,000. From the figure 3(b), the signature generation time gradually raises when the number of blocks increases, 2.10sec to 99.24sec.

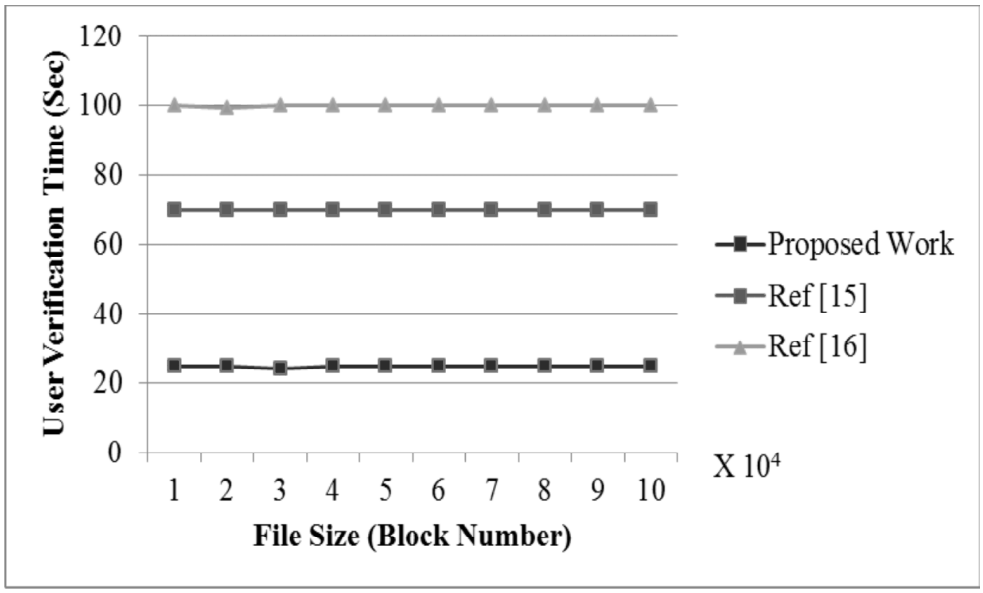

Figure 4: User Verification Time

Figure 4 shows the comparision results proposed work with other two algorithm in term of user verification time. from the results indicate that, although ref. [15] and [16] has comparable User Verification Time is very high when compared with proposed work.This is because ref. [15] [16]needs number of multiplication operations and exponentiation operationson Group number of challenge blocks, which are constant respectively.

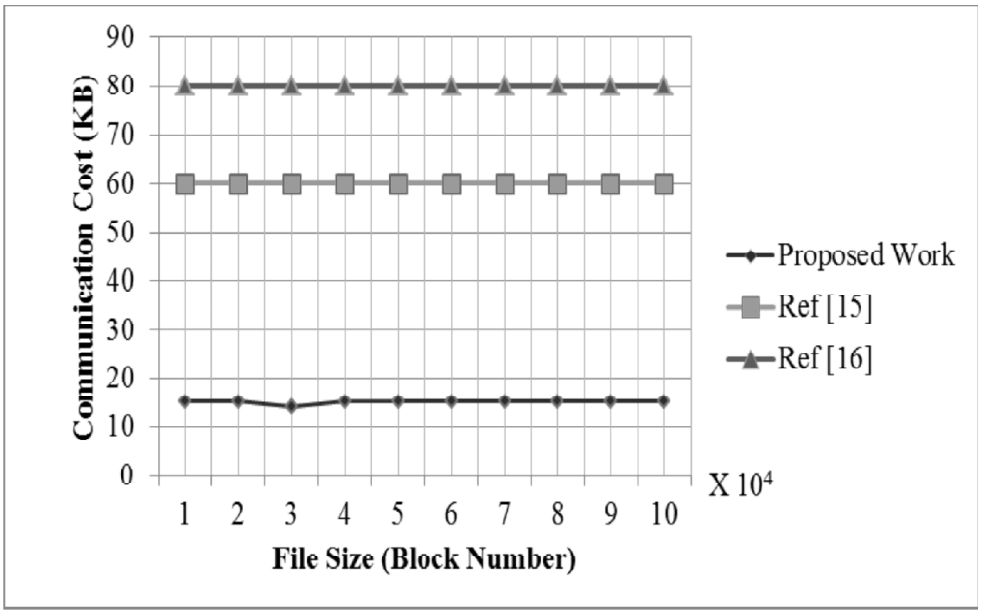

Figure 5: Communication Cost

Figure 5 shows the comparision results proposed work with other two algorithm in term of Communication Cost. from the results indicate that, although ref. [15] and [16] has comparable communication cost is very high when compared with proposed work and its computational cost also increases when communication cost increases on the TPA.

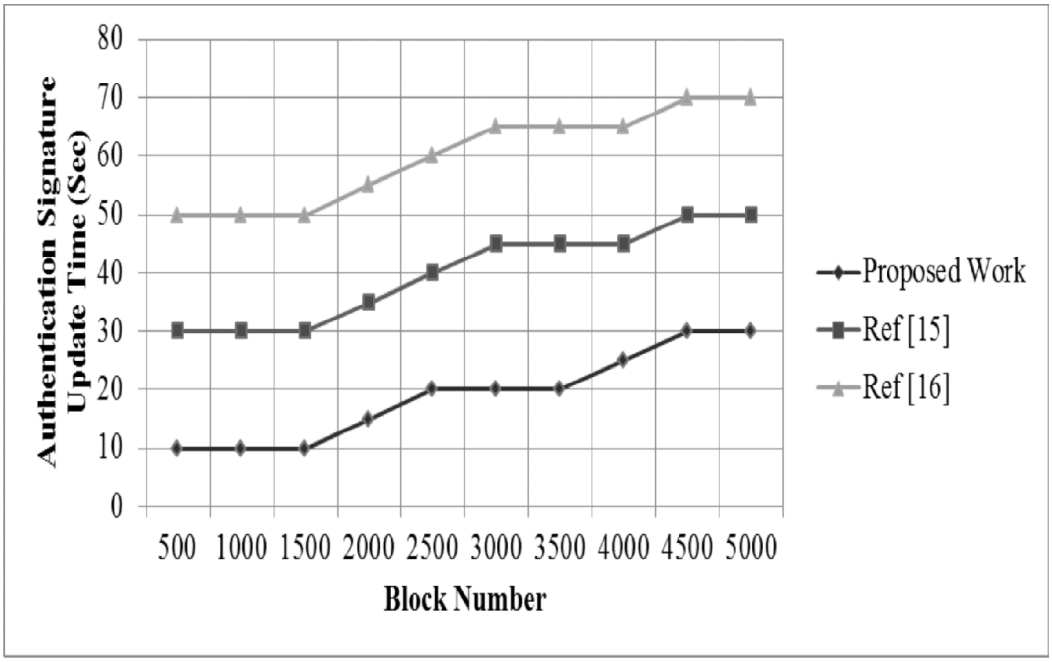

Figure 6: Authentication Signature Update Time

Figure 6 shows the comparision results of proposed work with other two algorithm in term of authentication signature update time. from the results indicate that, although ref. [15] and [16] has comparable authentication signature update time is very high when compared with proposed work. Thus, the proposed work shows the promising results when compared with other two process.

5. CONCLUSION

In this paper proposed a process to realize efficient and secure cloud auditing process of data integrity for share the dynamic data with group user with modification permission. A secret key sharing and a secret key authentication mechanism is used to manage the cloud storage data which is done with the help of the Homomorphic secret sharing and Salted Challenge Response Authentication Mechanism (SCRAM) techniques. Then the performance of the public auditing and data integrity is analyzed with the help of the experimental results. The experimental analysis shows the promising results in term of key generation timeauthentication signature generation time,user verification time,communication cost and authentication signature update time. Thus, proposed techniques authenticate the user and server secret keys while modifying the cloud file information that show the minimum error detection probability, efficient user revocation and public auditing.

REFERENCES

[1] Saranya Eswaran, Sunitha Abburu, "Identifying Data Integrity in the Cloud Storage", *IJCSI International Journal of Computer Science*, Issues, Vol. 9, Issue 2, No 1, PP. 403-408, 2012.

[2] Nedhal A. AL.Saiyd, Nada Sail, "Data Integrity In Cloud Computing Security ", *Journal of Theoretical and Applied Information Technology*, Vol. 58 No.3, PP. 570-581, 2013.

[3] Adarsh R,P. ChidanandaMurthy, "Secure Storage and Data Integrity Proof in Cloud", *International Journal of Computer Science and Information Technologies*, Vol. 5, No.4, PP. 5030-5032, 2014.

[4] Thombare Kishor V, SuryawanshiNilesh D, Patil Ganesh S, WadnereSwapnil, "Data Integrity Proof in Cloud Storage", *International Journal of Emerging Technology and Advanced Engineering*, Vol. 4, Issue 4, PP. 792-795, 2014.

[5] Siddhartha Rao, Savan Gujrathi, Mithun Sanghvi, Shubham Shah, "Analysis on Data Integrity in Cloud Environment", *IOSR Journal of Computer Engineering* (IOSR-JCE), Vol.16, Issue 5, Ver. IX, PP.71-76, 2014.

[6] M. Sowparnika, R. Dheenadayalu, "Improving data integrity on cloud storage services", *International Journal of Engineering Science Invention*, Vol.2, Issue 2, PP. 49-55, 2013.

[7] Anne Srijanya. K, N. Kasiviswanath, "Data Integrity Verification by Third Party Auditor in Remote Data Cloud", *International Journal of Soft Computing and Engineering* (IJSCE),Vol.3, Issue-5, PP.188-193, 2013.

[8] Cong Wang, Sherman S.-M.Chow, Qian Wang, KuiRen, Member, IEEE, and Wenjing Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage", *IEEE Transactions on Computers*, Vol. 62, No. 2, PP. 362-375, 2013.

[9] Alisha Jindal, Gagandeep, "Enhancing Data Integrity in Multi Cloud Storage", *International Journal of Engineering Research and Applications*, Vol. 4, Issue 9, Version 3, PP. 109-114, 2014.

[10] Jun Feng,Yu Chen, Wei-Shinn Ku, Pu Liu, "Analysis of Integrity Vulnerabilities and a Non-repudiation Protocol for Cloud Data Storage Platforms", *International Workshop on Security in Cloud Computing* (SCC), PP. 1-8, 2010.

[11] O. Rahamathunisa Begam, T. Manjula, T. Bharath Manohar, B. Susrutha, "Cooperative Schedule Data Possession for Integrity Verification in Multi-Cloud Storage", *International Journal of Modern Engineering Research* (IJMER),Vol. 3, Issue. 5, PP. 2726-274, 2013.

[12] Shlomi Dolev, Yin Li, "Secret Shared Random Access Machine", *https://eprint.iacr.org/2015/292.pdf*, 2015.

[13] Maha TEBAA, Saïd EL HAJJI, Abdellatif EL GHAZI, "Homomorphic Encryption Applied to the CloudComputing Security", *Proceedings of the World Congress on Engineering*, Vol.I, July 4 - 6, 2012.

[14] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," 16th IEEE Symp. Comput. Commun. (ISCC), Corfu, Greece, PP. 850–855, 2011.

[15] Pushkar Zagade, Shruti Yadav,Aishwarya Shah,Ravindra, "Group User Revocation and Integrity Auditing of Shared Data in Cloud Environment",International Journal of Computer Applications, Volume 128. No.12, PP.22-25, 2015.

[16] Tianzi Jiang , Xia Chen, Jiaxin Ma, "Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation", IEEE Transactions on Computers, Vol.1, Issue: 99, 2015.