PROVIDING DYNAMIC AUDIT SERVICES USING HOMOMORPHISM AUTHENTICATORS IN CLOUD INFRASTRUCTURE

[1]SATHIYA MOORTHY SRINIVASAN, [1]Dr. C CHANDRASEKAR,
[1]Research Scholar, Manonmaniam Sundaranar University, Tamil Nadu, India
[2]Asst. Professor, Peiryar University, Salem, India.
E-mail: [1]sathiyamoorthy.srinivasan@gmail.com, [2]ccsekar@gmail.com

ABSTRACT

Cloud storage enables client to access the data from anywhere at any time. Cloud based outsourced storage helps the clients to manage and conserve the data on self-sufficient platform. Probabilistic query and periodic verification on cloud zone provide public auditing but the cloud storages do not offer the clients with proof for verifying the integrity of the user. The authentication system based on Merkle Hash Tree construction is an efficient data dynamics system, yet not effective in improving storage. To improve the verification process on the cloud audit data services, Homomorphism Authenticators using Sphere Shaped Marker (HASSM) mechanism is proposed in this paper. HASSM mechanism develops a dynamic group and audits the integrity of stored data in the cloud services using the extendable dynamic hash structure. The sphere shaped marker signature verifies the information and subsequently audit the integrity of the shared data. With HASSM mechanism, the identity of the marker on each block stored data is kept private from a Third Party Auditor (TPA). TPA correctly detects any unauthorized users in HASSM for accessing the cloud files by applying KeyGener, SphereSign and SphereVerify algorithm to perform the process. TPA uses HASSM mechanism to validate the integrity of stored information without retrieving the whole file, so that the computational cost is reduced. HASSM Mechanism provides the approximately 9.188 % better verification result with lesser computational cost. Experiment is conducted in the Cloudsim platform on the factors such as auditing time, read/write efficiency on cloud services, and bandwidth utilization for data auditing.

Keywords: *Data Homomorphism Authenticators, Cloud Storage, Sphere Shaped Marker, Data Service, SphereSign, Integrity, SphereVerify*

1. INTRODUCTION

In recent years the cloud storage services have receiver greater attention due to the faster growth rate. The faster growth rate denotes the scalability, lesser cost to build the cloud infrastructure for business applications and so on. The cloud infrastructure is build on the basis of open architecture based interface. The cloud zone is capable to incorporate the requirements of multiple internal and external cloud users to improve the interoperability. Cloud computing at present act as most popular technologies with the ability to reduce the cost associated with secure computing. In a cloud computing environment, if a device is recorded with a particular cloud service provider, then the server authenticate each other in a consistent way in order to secure the user communication.

The success of cloud service depends on the effective use of resources by the end users. Cloud computing service is convenient for file storage and also reduces the storage capacity on individual system. The impact of cloud computing on digital forensics as described in [6] analyzes all type of digital forensics. Digital forensics finds the criminals through the overflow technology but flat corporate network and the social engineering path was not achieved.

Cloud computing is very suitable to access the files from the mutual group of configurable computing resources like server, network and resources. The user communication is provided by auditing the service using single authentication technique. Currently, cloud user validation is performed using different methods namely password authentication, Graphical and 3D password, and signature based authentication.

A group of cloud server architecture is used in cloud computing for performing the user authentication. The most popular user authentication scheme is the user id, password method for the verification. The password is generated using the hash function in the existing system. Smartcard also used in the cloud zone to prevent the unauthorized user attacks. Cloud User authentication is also provided with graphical password authentication and 3D password objects but it consumed more time and space during processing. Some cloud infrastructure systems have constructed authentication system based on sending the Short Message Service (SMS), but it doesn't promise a SMS delivery within the time.

A unique compression technique as illustrated in [5] maintained clinical reference and balanced the legal risk. The medical images stored on the cloud reduce the space consumption but the cloud panel was not improved with the Graphical User Interface (GUI). The ciphered channeling algorithm did not assured security. File Assured Deletion (FADE) built a set of cryptographic key function in [12] for maintaining the cloud information with high security percentage. The quorum key manager's acts as an overlie scheme which included value in addition to security features for today's cloud storage services.

Trustworthy Resource Scheduling in Clouds as described in [4] assured the user with virtual resources. The resources were hosted using physical services which were equivalent to their requirements without getting users involved in understanding the details of the cloud infrastructure. But resource scheduling failed in establishing trustworthy collection and calculation of the other properties. Wireless Link Scheduling for Data Center Networks (WLS-DCN) in [13] balanced user request query traffic and

maximized the network utility. The network utility with limited wireless resources and co-channel interference confirmed the performance. But WLS-DCN failed to perform join operation (i.e.,) to combine the joint operation in order to minimize the utilization of wireless resources.

Collaborative Provable Data Possession (PDP) scheme agrees to the techniques of homomorphism in [8] with hash index hierarchy to perform join operation. The hash index hierarchy produced the tags with the additional time taken when compared to the size of data blocks and also not effective on practical CPDP constructions. Security-Mediator (SEM) as presented in [10] generated the verification metadata (i.e., signatures) on outsourced data for data owners and decoupled the secrecy safety mechanism from Provable Data Possession (PDP). PDP not only minimized the computation and bandwidth requirement of this mediator, but also reduced the trust located on it in terms of data privacy and individuality privacy.

Communication protocol accommodated non-repudiation of origin in [20] and privacy provided on message originator. Unforgivable digital signatures were provided with special kind of proof. Condor based process virtualization platform as presented in [9] used the resources to process, direct and deal with huge quantity of information from the cloud environment. Internet-based paradigms solved the expanding data and task concerted computation problems. But the virtualization platform failed to integrate the strategies into parallel distributed form for auditing the data with minimal time per task.

Two Way Integrity (TWI) algorithms included TPA as illustrated in [18] to check the reliability of the cloud system. But TWI failed to present effective data integrity check between the third party auditor and the cloud service provider. TWI also leaked the data from the cloud service provider. Secure cloud storage system support public auditing with high privacy TPA. TPA as demonstrated in [17] permitted an exterior auditor to audit the user's cloud information before the data substance was produced to the users. Batch auditing with multiple delegated auditing tasks were performed simultaneously on different users but robustly did not coped up with very large scale data.

Data integrity in the cloud as explained in [19] employed to check the correctness of data in the cloud on behalf of the user. For this specific purpose, Service Level Agreement (SLA) was used for proof checking with the cloud and the customer. The network bandwidth was wasted on large extent for less bit of information. RSA scheme was applied only to static storage of data, whereas the dynamic operation was not performed.

Multi-server data comparison algorithm as demonstrated in [11] updated the information in the cloud server. Multi-server data comparison efficiently checked the integrity and also the security parameter was analyzed. Multi-server data comparison does not execute the security protocol in cloud data storage for auditing. An un-trusted and outsourced storage created an Interactive Proof System (IPS) in [1] and performed dynamic audit service for verifying the integrity. Audit service provided public audit ability without downloading raw data but still the cloud storages should offer clients with a further significant proof checking mechanism for integrity verification for the data stored in order to minimize the cost involved during communication.

In the proposed work, Homomorphism Authenticators using Sphere Shaped Marker (HASSM) mechanism is developed to authenticate and provide the cloud data accessing ability to the end users. Sphere Shaped Marker signature is introduced to audit the data without downloading the files. Due to this the computational cost is reduced to great extent using HASSM Mechanism, as the downloading of file is avoided during the verification. The HASSM includes KeyGener, SphereSign and SphereVerify for verification of single and group users where the users vary dynamically in the group and in addition the users are authenticated effectively using TPA.

The structure of this paper is as follows. In Section 1, the basic problems involved during data auditing using cloud infrastructure is presented. In Section 2, a mechanism named Homomorphism Authenticators using Sphere Shaped Marker is designed. Section 3 outline experiment results with cloud simulator. Section 4 produces the results. Finally, Section 5 demonstrates the related work and Section 6 concludes the work with effective data authentication system in cloud zone.

2. HOMOMORPHISM AUTHENTICATORS USING SPHERE SHAPED MARKER MECHANISM

The main objective of the homomorphism authenticators is to audit the shared data on the cloud zone. The authentication is essential while transferring the confidential data over the cloud data services. The users in the cloud zone ranges from single original user to the group users. In HASSM mechanism, a dynamic group is developed where a new user is added into the group and an existing group member is revoked during data sharing in cloud. The sphere shaped marker signature authentication is used to verify with the Third Party Auditor (TPA). The sphere shaped signature is introduced in HASSM mechanism to hide the identity of the signer on each block. Flow diagram of HASSM Mechanism is described in Fig 1.

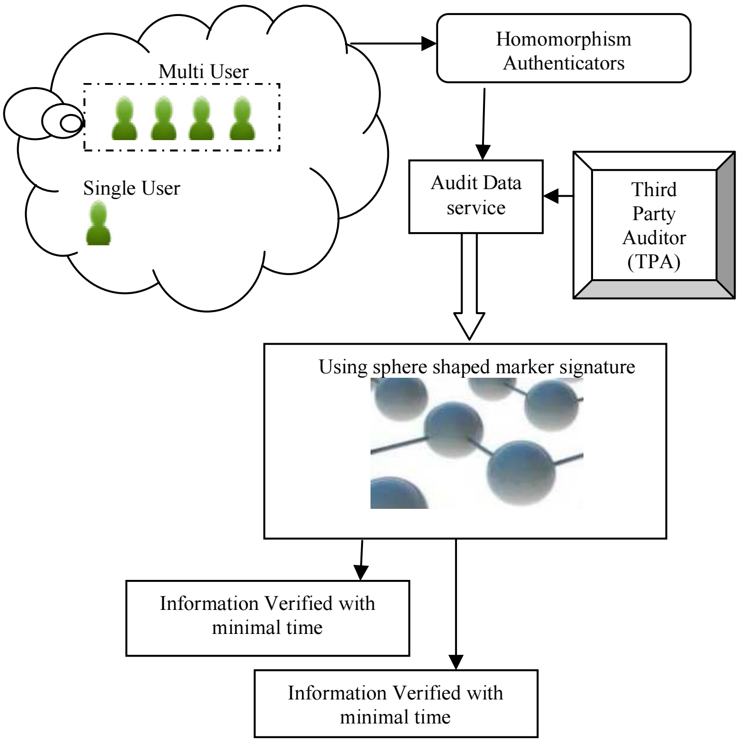
Multi User
Single User
Homomorphism Authenticators
Audit Data service
Third Party Auditor (TPA)
Using sphere shaped marker signature
Information Verified with minimal time
Information Verified with minimal time

Figure 1 Flow Diagram of HASSM mechanism

As illustrated in Figure 1, HASSM mechanism consists of the cloud zone which handles both the single and group users. Whenever a single user requests the cloud zone for accessing the stored information, the request is sent to the Third Party Auditor (TPA). The TAP verifies the integrity of shared data in the cloud zone using Homomorphism Authenticators with sphere signature. Moreover, if a single or group users need to fetch the information from the cloud network zone, then the verification is carried out using sphere shaped marker signature before sharing the data. The job of sphere shaped marker signature is to match with the stored signature, and then the access is provided by TPA with auditing report so that the data is shared efficiently.

Sphere shaped marker signature is employed in the HASSM mechanism to cover the identity of the signer on each block. The sphere shaped is covered with the block so that the files are verified in the TPA zone without downloading the file. The avoidance of downloading the file reduces the audit time per task and also reduces the computational cost. The sphere signatures in homomorphism authenticators maintain the signatures and provide the effective identity on auditing the data in cloud.

The forthcoming subsection discusses in detail about the steps involved during the construction of homomorphism mechanism which is followed by the design of homomorphism authenticators using sphere shaped marker and finally the steps involved during the third party auditing for effective verification process.

2.1 Homomorphism Construction

Homomorphism authentication involves the process of authenticating the users (i.e.,) single or groups of users before the data are shared from the cloud zone. Homomorphism authenticators perform effective mathematical operations to authenticate the users even on dynamic structure. Homomorphism describes the transformation of one data file to another while conserving relationships between elements but their structure remains the same during the authentication process.

Homomorphism authenticators perform the verification and audit the data service using the HASSM mechanism. The construction of homomorphism follows in such a way that a user with a private key generates valid signatures where the sphere signer's private and public key is denoted as $(K_{private}, K_{public})$. Sphere Signatures are denoted as $\delta_1, \delta_2, \delta_3 \dots \delta_n$ and blocks are represented as $b_1, b_2, b_3 \dots b_n$. The overall block structure with sphere signature using HASSM mechanism is denoted as given below:

$$B' = \delta_1 b_1 + \delta_2 b_2 + \delta_3 b_3 + \cdots. + \delta_n b_n$$

……..Eqn (1)

Eqn (1) denotes the complete block with the sphere signatures denoted in B'. The sphere signature is covered inside each block in HASSM mechanism till the b_n blocks are arrived. Block verification allows a TPA to easily verify and produce the audit result. The block based Sphere Signature covering is shown in the Figure 2.

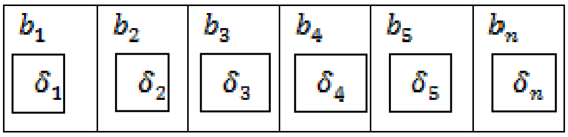

Figure 2 Representation of Sphere Signature in Block

Block verification with sphere signature in HASSM allows a verifier to audit the correctness of data stored in the cloud zone. The block verification is carried out using Dynamic hashing function in HASSSM mechanism. In order to carry out extendable dynamic hash function, two blocks are combined together to join the sphere signatures. HASSSM mechanism uses the dynamic hashing with 'n' bit integers to handle large group of dynamic users. The dynamic users take 'n' bit with the hash function to index the table of bucket addresses. The value of 'n' increases and decreases, as the number of users are included or excluded from HASSSM cloud zone. The extendable hash structure combines the sphere signature in the blocks. The extendable hash function is shown in Figure 3.

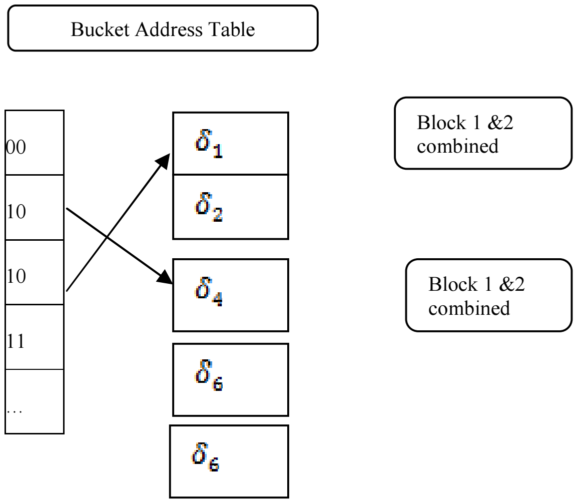

Figure 3 Extendable Hash Structure based Block Combining

Extendable hash structure in HASSM combines the multiple blocks with sphere signature. The bucket address table contains all the addresses of both the single and group users. As the users change dynamically, the bucket address table is also modified automatically in HASSM mechanism. The extendable hash function for combining of the block is formularized as,

$$EHF: \{0,1\} \rightarrow \{b_4, b_6, b_8\} \quad \text{……..Eqn (2)}$$

$$EHF: \{1,0\} \rightarrow \{b_1, b_2\} \quad \text{…….. Eqn (3)}$$

EHF is the extendable hash function with bucket address table. The bucket address table contains the combined block signatures. The extendable hash function is computed to combine the sphere signature block in cloud zone.

2.2 Homomorphism Authenticators using Sphere Shaped Marker

Once the sphere signature blocks are combined in the cloud zone, Sphere Shaped Marker authenticates the cloud users before the data are shared. The auditing of data is carried out by the Third Party Auditor (TPA) using block type verification. The HASSM consists of KeyGener, SphereSign and SphereVerify algorithm to perform the processing in cloud zone. In HASSM each user in the dynamic group generates the private $\mathrm{K}_{\mathrm{private}}$ and public $\mathrm{K}_{\mathrm{public}}$ key using KeyGener,. In HASSM, SphereSign helps both the single and group user to sign in a block using the private $\mathrm{K}_{\mathrm{private}}$ key. Using HASSM mechanism, SphereVerify check whether the given user sign is matched with the block sign for data sharing or not. Let us assume that $\mathrm{M}_1, \mathrm{M}_2, \dots \mathrm{M}_n$ are the multiplicative sphere groups in cloud zone for auditing the data and let m_1 and m_2 be the generators for $\mathrm{M}_1, \mathrm{M}_2$ mapping.

$$e: \mathrm{M}_1 * \mathrm{M}_2 \rightarrow \mathrm{M}_n \quad \dots\dots\dots \text{Eqn (4)}$$

Where M_1 and M_2 denotes the linear mapping group with the computable homomorphism. The homomorphism maintains the same structure to audit the data from TPA but the signature marker is changed in HASSM as sphere signature marker for reducing the communication cost is introduced. The total number of dynamic users in the group is denoted as 'D' KeyGener, SphereSign and SphereVerify algorithm in HASSM mechanism is described as,

```
// HASSM KeyGener

Step 1: User 'U' randomly picks the key to select
the private key $K_{private}$, public key$K_{public}$
Step 2: Public key = {Weight 'W' of all 'U'} keys
Step 3: Private Key = Random Key generation by
user

// HASSM SphereSign

Step 4: 'D' users' public key used to identify the
Block ID
Step 5: Private key  randomly chosen by User 'U'
is computed as
        $SS = EHF\,(B_{id})m_1 \in M_1$
Step 6: Sphere Signature on Block 'B' contain
$\delta = (\delta_1, \delta_2, \delta_3 \dots \delta_n) \in M_1^D$
```

```
//HASSM SphereVerify

Step 7: D' users public key verified with Block ID
Step 8: Sphere Signature verified with
SphereVerify operation
Step 9: $e(SS, M_2) = \prod_{i=1}^{D} e(\delta_n, W_n)$ verify the
keys
Step 10:  If key satisfied above equation, then block
'B' sign is matched with user sign
Step 11: Then access provided to user for cloud
data usage
Step 12: Else access not provided to the user
```

The above algorithm initially generates the key in HASSM. Both the public and private key is generated by the users. The public key is created based on the weight of the users in the cloud infrastructure. The private key is chosen randomly using the HASSM KeyGener. The SphereSign uses the 'D' users to identify the Block ID on which the signature is covered. The covered signature is a sphere sign signature of 'U' users and computed in step 5. The computed sphere signature on block 'B' is verified in the SphereVerify Algorithm. Public Key is verified with Block ID and step 9 is significantly used to verify the keys. The key verification helps to authenticate the users and provide access for the data sharing.

2.3 Third Party Auditing in HASSM

Once the key verification is accomplished, the Third Party Auditing (TPA) performs the effective verification process for every user (i.e.,) single or group user request. TPA view the data blocks with sphere signature to clearly audit the task with minimal auditing time. The auditing flow diagram for TPA is shown in Figure 4.

TPA auditing flow diagram illustrated using HASSM mechanism in cloud infrastructure is demonstrated in figure 4. TPA correctly detects the presence of any unauthorized users is trying to access the cloud files using HASSM. If the user request satisfies the KeyGener, then TPA view the block ID. The Block ID checks the sphere signature using the SphereVerify Algorithm. If the Sphere signature matches the stored signature block, then the HASSM provide access to view the files for the end users. If the Sphere signature is not matched with the stored signature block, then the next Block is viewed, till then B_n block search process is completed.

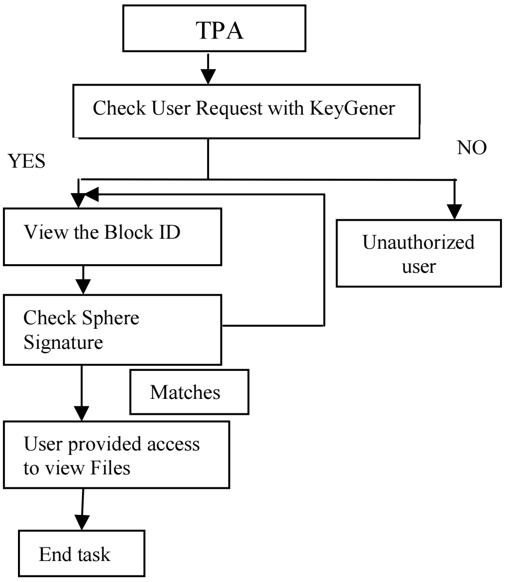

Figure 4 TPA auditing Flow Diagram

3. EXPERIMENTAL EVALUATION OF HASSM MECHANISM

Homomorphism Authenticators using Sphere Shaped Marker (HASSM) mechanism performs the experimental work using Java platform. The JAVA CloudSim simulator is introduced to evaluate the data audit services. The specified toolkit has been selected as a simulation platform as it is a present simulation structure in Cloud computing environments. Compared to the simulation toolkits (e.g. SimGrid, CloudSim), it provides copy of on-demand virtualization enabled bandwidth and data service management. To demonstrate the experimental work on the cloud simulator 8 GB of RAM and 1 TB of storage space is required.

Record Linkage Comparison Patterns Data Set from UCI repository is extracted for the evaluation of the HASSM Mechanism. The task compares the pattern and audit (i.e.,) verify whether the underlying records belong to that person or not and access the information accordingly. HASSM Mechanism compares the experimental result against the Probabilistic query and Periodic Verification (PPV) method and Merkle Hash Tree (MHT) construction method in terms of audit time per task, User Queried rate, computational cost, bandwidth utilization, read/write efficiency on cloud services and data verification rate. Audit time per task is defined as the amount of time taken to perform the auditing operation by the TPA. Audit time is measured in terms of milliseconds (ms) and it is defined as,

[Audit Time per Task = Audit Start Time - Audit End Time]

User queried rate denotes the amount of effectiveness in producing the result to the end users. Query rate at which the request files are addressed to the users from cloud data center. User queried rate is measured in terms of percentage (%). The Computational cost is the mathematical working out of resources to attain the solution for the user problem.

$$\text{Computational Cost} = \text{File Size} * \text{Cost per unit}$$

The computational cost is measured according to the size of file and the cost incurred for each unit and the overall cost parameter is denoted in terms of Kilo Bytes (KB). Bandwidth utilization is defined as the bit rate variable consumed to perform the data communication in the cloud infrastructure. The utilization of bandwidth is higher in HASSM mechanism and it is measured in terms of Mega bits per second (Mbit/s).

$$\text{Bandwidth Utilization} = \text{Block Size} * [100 \text{ bits per second}]$$

Bandwidth utilization is used to easily analyze the data audit rate of each user. The higher the data audit rate, the lower the auditing time will be. The effectiveness on read and write operation on the cloud infrastructure is defined as the read/write efficiency on cloud services. The read/write efficiency is measured in terms of percentage (%). Data verification rate is carried out in the TPA using the SphereVerify algorithm, measured in terms of % per task. The verification rate is improved using the sphere signature block in HASSM mechanism.

4. RESULT ANALYSIS OF HASSM

In section 4, HASSM Mechanism results are analyzed with the existing Probabilistic query and Periodic Verification (PPV) method and Merkle Hash Tree (MHT) construction method. The existing and proposed result is analyzed through table values and graph points.

Table 1 Tabulation of Audit Time per Task

No .of Audit tasks	Audit Time per Task (ms)		
	PPV Method	MHT method	HASSM Mechanism
50	440	420	400
100	910	880	850
150	1200	1182	1145
200	1530	1495	1462
250	1925	1885	1724
300	2099	2056	2012
350	2350	2298	2256

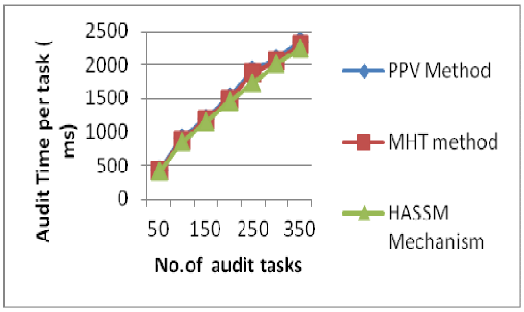

Figure 5 Performance of Audit Time per Task

Figure 5 illustrate the audit time per task based on the task count. As the user request task gets increased, the audit time is reduced using HASSM mechanism when compared to the existing method. A user with a private key generates a valid signature using the HASSM properties and placed in the blocks. With this the audit time get reduced from 4 – 10 % when compared to the PPV Method [1] due to file block usage and gets reduced from 2 – 8 % when compared to the MHT method [2].

Table 2 Tabulation of User Queried Rate

No. of File Blocks	User Queried Rate (%)		
	PPV Method	MHT method	HASSM Mechanism
20	65	72	80
40	66	74	82
600	67	75	85
80	69	78	87
100	70	80	88
120	72	81	90
140	74	82	92

Table 2 describes the user queried rate based on the file blocks. As file blocks size gets increased, the queried rate is also improved.

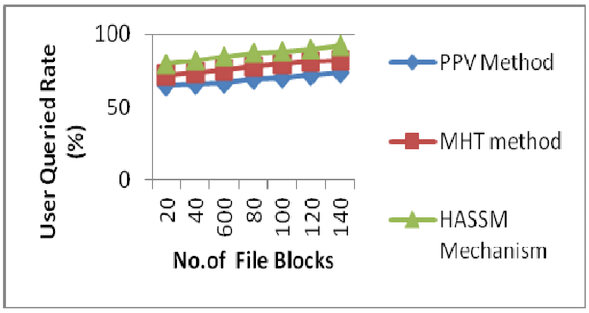

Figure 6 Measure of User Queried Rate

Figure 6 describes the user queried rate using HASSM mechanism and compared in an elaborate manner with the existing methods. Block verification with sphere signature in HASSM allows a verifier to audit the correctness with improved user query rate. The block verification is carried out using the Dynamic Extendable hashing function and improves the query rate by 23- 26 % when compared with the PPV Method [1]. The user queried rate is also improved by 10 – 13 % when compared with the MHT method [2].

3 Tabulation of Computational Cost

File Size (KB)	Computational Cost (KB)		
	PPV Method	MHT method	HASSM Mechanism
1000	3495	3350	2998
2000	6601	6556	6001
3000	9798	9535	8895
4000	12650	12580	12000
5000	15700	15690	14982
6000	18818	18760	18010
7000	21970	21790	21020

Figure 7 demonstrates the computational cost based on the file size. The sphere shaped is covered with the block and the verification carried out by TPA without downloading the file. The avoidance of file download reduces the computational cost from 3 – 10 % in HASSM when compared with the PPV Method [1]. The sphere signatures in homomorphism authenticators maintain the signatures and reduce the computational cost by 4 – 14 % when compared with the MHT method [2].

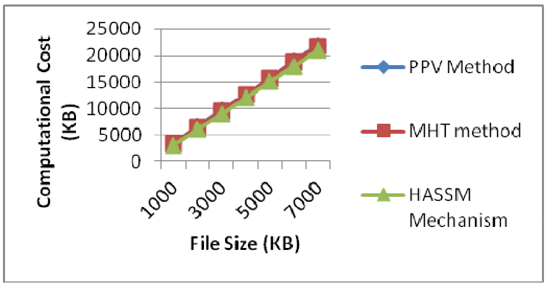

Figure 7 Measure of Computational Cost

Table 4 Tabulation of Bandwidth Utilization

Block Size (KB)	Bandwidth Utilization (M bits/s)		
	PPV Method	MHT method	HASSM Mechanism
250	23150	24100	25000
500	44250	46000	49989
750	69650	70010	74990
1000	94450	96150	100010
1250	119780	121100	125020
1500	141220	145000	149880
1750	167400	169006	175025

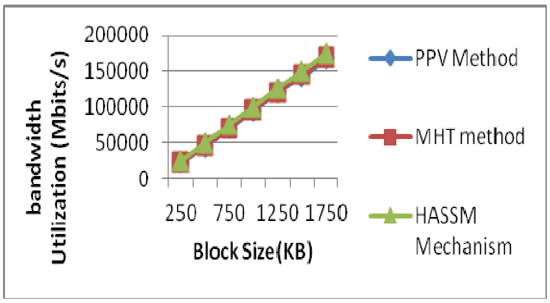

Figure 8 Measure of Bandwidth Utilization

Figure 8 presented the result graph of bandwidth utilization. Bandwidth utilization is used to measure the data audit rate. The extendable hash structure in HASSM combines multiple blocks using sphere signature. The bucket address table contains all the addresses of both the single and group users so that the bandwidth utilization is improved by 4 -12 % when compared with the PPV Method [1]. The bucket address table is also modified automatically based on dynamic users and improves the bandwidth rate by 3 – 8 % when compared with the MHT method [2].

Table 5 Tabulation of Read/Write Efficiency on Cloud Services

User count	Read/Write Efficiency on Cloud Services (%)		
	PPV Method	MHT method	HASSM Mechanism
10	70	72	80
20	71	73	81
30	73	75	82
40	74	76	85
50	75	78	87
60	77	80	89
70	79	82	94

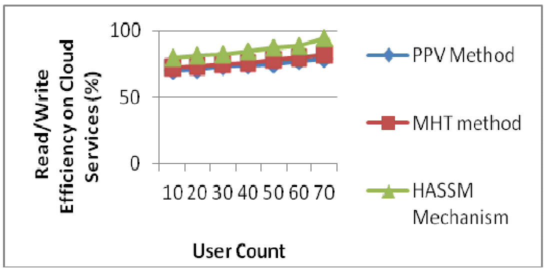

Figure 9 Measure of Read/Write Efficiency on Cloud Services Measure

Figure 9 describes the read/write efficiency on the cloud services based on the user count. Homomorphism authenticators perform the effective mathematical operations to authenticate the users and improve the read/write efficiency rate. The read/write efficiency rate is 12 – 18 % improved when compared with the PPV Method [1]. The homomorphism is described with the same structure, so that the read and write operation is made easier with improved result from 9 – 14 % using HASSM mechanism when compared with the MHT method [2].

Table 6 Data Verification Rate Tabulation

Task	Data Verification Rate (% per task)		
	PPV Method	MHT method	HASSM Mechanism
Task_1	80.5	82.5	90.5
Task_2	81.6	83.6	91.2
Task_3	82.5	84.9	93.5
Task_4	84.9	85	94
Task_5	85.2	87	95.6
Task_6	87.6	89.1	97
Task_7	90.1	92.4	98

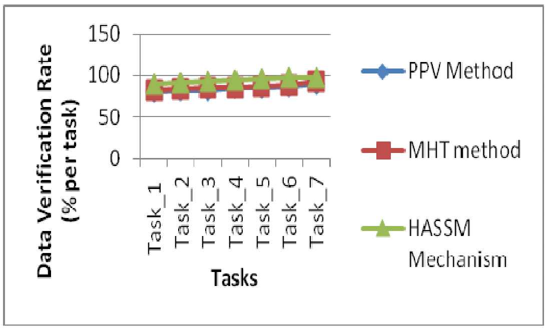

Figure 10 Data Verification Rate Measure

Figure 10 presented data verification rate with different user's task. Both the single and group user's signatures are verified with block signature to fetch the information from the cloud network zone. The sphere shaped marker signature is matched with the stored signature, and as a result the verification rate is improved from 8 – 13 % when compared with the PPV Method [1]. SphereVerify check the given user sign with the block sign for data sharing and improves the data verification rate from 6 – 10 % when compared with the MHT method [2].

Finally, Homomorphism Authenticators using Sphere Shaped Marker developed an authentication system on dynamic group users and audits the data services without the file, so that the computational cost is reduced. The HASSM algorithm is used widely to audit the integrity of the shared data.

5. RELATED WORK

Provider-aware anonymization algorithm as illustrated in [14] with adaptive m-privacy examination policy mad sure the high usefulness and m-privacy of anonymized data with efficiency. Anonymization presented heuristic algorithm for resourceful confirmation and exploited the equivalence group monotonicity property. But the lookahead approach conspired the situation with m-privacy. Secure and Privacy conserve Keyword Searching scheme (SPKS) as presented in [16] allowed cloud service provider to contribute towards decipherment. On the other hand, CSP searches the encrypted file efficiently without leaking any information. Public key based homomorphism linear authenticator as demonstrated in [3] facilitates TPA to execute the auditing without requesting the neighboring copy of information. Public auditability for cloud storage is of serious significance so that users resort a TPA for outsourced data. The outsourced data are fault free but computational expenditure of cloud audit services is high. TPA using Boneh–Lynn–Shacham (BLS) algorithm for signing in [7] confirmed the integrity of the data stored in the cloud and resource allocation. External TPA, on behalf of the cloud user confirm the reliability of the data stored in the cloud by utilizing public key based homomorphism authenticator with accidental masking privacy preserving public auditing. The most capable model with public verifiability is enforced and does not permit TPA to audit the cloud data storage which is devoid of challenging users' time and resources.

Existing Public Auditability and Data Dynamics for Storage Security designed in [2] a verification scheme for faultless integration of features. Classic Merkle Hash Tree construction accomplished resourceful data dynamics, but yet not effective in improving the storage models by directing the classic Merkle Hash Tree construction for block tag authentication. Dynamic challenger attacks as demonstrated with three auditing mechanisms in [15] split data in the cloud. The data shared with the privacy-preserving auditing mechanisms called Oruta and Knox constructed hash tree. A distributed storage integrity auditing mechanism arbitrarily altered the cloud data during the verification phase, but consumed more cost.

6. CONCULTION

Homomorphism Authenticators provide the effective authentication with TPA while sharing the files to the end users using the cloud infrastructure. Sphere Shaped Marker is developed to handle the dynamic set of group users in the cloud zone and to verify the signature information and audit the integrity of the shared data. Moreover, TPA correctly detects any unauthorized users in HASSM using KeyGener, SphereSign and SphereVerify algorithm for accessing the cloud files. Block verification allows a TPA to easily verify the HASSM mechanism and produce the audit result with higher percentage result. The extendable dynamic hash structure helps to combine the sphere signature in the blocks of HASSM mechanism. HASSM mechanism audits the users and verifies the information without retrieving the whole file, so that the computational cost is reduced to 10 % when compared to the state-of-art method. Cloudsim simulator result provide averagely 6.185 % minimal auditing time and provide effective cloud services on the read and write operations.

REFRENCES:

[1] Yan Zhu., Gail-Joon Ahn., Hongxin Hu., Stephen S. Yau., Ho G. An., and Chang-Jun Hu., "Dynamic Audit Services for Outsourced Storages in Clouds," IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 6, NO. 2, APRIL-JUNE 2013

[2] Qian Wang., Cong Wang., Kui Ren., Wenjing Lou., and Jin Li., "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 22, NO. 5, MAY 2011

[3] Cong Wang., Sherman S.-M. Chow, Qian Wang., Kui Ren., and Wenjing Lou., "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE TRANSACTIONS ON CLOUD COMPUTING YEAR 2013

[4] Imad M. Abbadi., and Anbang Ruan., "Towards Trustworthy Resource Scheduling in Clouds," IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 8, NO. 6, JUNE 2013

[5] S.S.Viswa., "Efficient Storage of Medical Image on Cloud," International Conference on Emerging research on Commuting, Information, Communication and Applications, 2013

[6] Farid Daryabar., Ali Dehghantanha., Nur Izura Udzir., Nor Fazlida binti Mohd Sani., Solahuddin bin Shamsuddin., Farhood Norouzizadeh., "A Survey About Impacts of Cloud Computing on Digital Forensics," International Journal of Cyber-Security and Digital Forensics (IJCSDF), The Society of Digital Information and Wireless Communications, 2013

[7] JACHAK K.B., KORDE S.K., GHORPADE P.P. AND GAGARE G.J., "HOMOMORPHIC AUTHENTICATION WITH RANDOM MASKING TECHNIQUE ENSURING PRI-VACY & SECURITY IN CLOUD COMPUTING, "BIOINFO Security Informatics ISSN: 2249-9423 & E-ISSN: 2249-9431, Volume 2, Issue 2, 2012

[8] YAN ZHU., and SHANBIAO WANG., "SECURE COLLABORATIVE INTEGRITY VERIFICATION FOR HYBRID CLOUD ENVIRONMENTS," International Journal of Cooperative Information Systems Vol. 21, No. 3 165–197. DOI: 10.1142/S0218843012410018, (2012)

[9] Haiyan Guan., Jonathan Li., Liang Zhong., Yongtao Yub., Michael Chapman., "Process virtualization of large-scale lidar data in a cloud computing environment," Computers & Geosciences., Elsevier Journal., 2013

[10] Byang Wang., Sherman S. M. Chow., Ming Li§, and Hui Li., "Storing Shared Data on the Cloud via Security-Mediator," ACM CCSW, 2011

[11]Dinesh.C., "Data Integrity and Dynamic Storage Way in Cloud Computing," Distributed, Parallel, and Cluster Computing, arXiv.org > cs > arXiv:1111.2418., 2011

[12] Yag Tang., Patrick P. C. Lee., John C. S. Lui., Radia Perlman., "Secure Overlay Cloud Storage with Access Control and Assured Deletion," IEEE TRANSACTIONS ON DEPEDABLE AND SECURE COMPUTING., 2012

[13]Yong Cui., Hongyi Wang., Xiuzhen Cheng., "Wireless Link Scheduling for Data Center Networks," ACM journal., 2011

[14 Slawomir Goryczka., Li Xiong., and Benjamin C. M. Fung., "m-Privacy for Collaborative Data Publishing," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2013

[15] Yng Yu., Lei Niu., Guomin Yang., Yi Mu., Willy Susilo., "On the security of auditing mechanisms for secure cloud storage," Future Generation Computer Systems: the international journal of grid computing., 2014

[16] Qn Liu., GuojunWang., JieWub., "Secure and privacy preserving keyword searching for cloud storage services," Journal of Network and Computer Applications., Elsevier Journal.,2011

[17] Cog Wang., Sherman S.M. Chow, Qian Wang., Kui Ren., and Wenjing Lou., "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Transactions on (Volume:62 , Issue: 2) Computers, 2013

[18] Garima., "Ensuring Data Storage Security in Cloud Using Two Way Integrity Check Algorithm," International Journal of Advanced Research in Computer Science and Software

Engineering., Volume 3, Issue 11, November 2013

[19] S.K. Prashanth., Dr N. Sambasiva Rao., K. SUNITHA., V. TEJASWINI., "Enabling Integrity for the Compressed Files in Cloud Server," IOSR Journal of Computer Engineering (IOSR-JCE) Volume 12, Issue 4, 2013

[20] Wei Wu., Jianying Zhou., Yang Xiangc., Li Xua., "How to achieve non-repudiation of origin with privacy protection in cloud computing," Journal of Computer and System Sciences., Elsevier journal, 2013