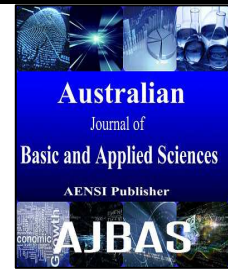




## AUSTRALIAN JOURNAL OF BASIC AND APPLIED SCIENCES

ISSN:1991-8178 EISSN: 2309-8414  
Journal home page: www.ajbasweb.com



### Ensuring Cloud Security and Privacy Through Enhanced Merkle Hash Tree Method Using Bat Mechanism

<sup>1</sup>S. Srinivasan and <sup>2</sup>K. Raja

<sup>1</sup>Research Scholar, Research Development center, Bharathiar University, Associate Professor, Department of M.C.A,K.C.G College of Technology, India.

<sup>2</sup>Principal and Dean Academics, Alpha College of Engineering, India,

#### Address For Correspondence:

S. Srinivasan, Research Scholar, Research Development center, Bharathiar University, Associate Professor, Department of M.C.A,K.C.G College of Technology, India.  
E-mail: effectivemail@yahoo.com

#### ARTICLE INFO

##### Article history:

Received 12 February 2016

Accepted 12 March 2016

Available online 20 March 2016

##### Keywords:

Authentication, cloud security, data confidentiality, dynamic data integrity, public auditability.

#### ABSTRACT

Cloud computing delivers proper on-demand broad range of network access, services and resources like computational utilities, storage and applications to cloud consumers through internet by pay-per-usage basics. With an rising number of cloud providers resorting to exploit and distribution of computing resources and services in cloud computing, there is a need for defending the confidential information of different users from malicious users and attackers. However, integrity of data storage, security and users privacy of an open ended, sensibly division of available computing services and resources is still ambiguity and present a major difficulty for cloud consumers to adapt in cloud environment. This paper deals cloud security concern includes dynamic data integrity and public auditability. This paper deals the protection concern includes many of cloud attacks, data leakage, privacy, confidentiality, threats while giving out of resources and services. This method deals securing the cloud information, data authentication, users' data auditability, data integrity through enhanced merkle hash tree construction mechanism using bat algorithm. The bat algorithm is a meta-heuristic algorithm stimulated by the echolocation behavior of micro-bats used to optimize the encrypted values in enhance merkle hash tree technique. This method verifies the difficulty of public auditability can achieved through multiple auditing tasks simultaneously by third party auditor in cloud environment and make sure of confirm the vibrant data integrity kept in cloud. This capable method conserve the cloud computing environment with better performance evaluation. Furthermore, cloud security, data integrity and privacy analysis knows the facility of the proposed approach for cloud environment and extent dynamic efficiency with secure cloud computing environments.

#### I. Introduction To Cloud Computing:

Currently cloud computing is a quickly developing increasing internet-based novel information technology of distributed computing of computer industry (Lin, G., 2012). Cloud environment provide technology enabled services, shared resources, software, hardware and platforms to the cloud clients and organizations. Cloud computing is based on several attributes such as multi-tenancy, elasticity, scalability, pay-as-you-go and self provisioning of resources (Mohamed, E.M., 2012). Cloud providers have Infrastructure as a Service, Platform as a Service, Software as Service and other services to present. The cloud deployment model provides public, private, community and hybrid cloud (Behl, A., K. Behl, 2012). Cloud computing facilitates to decrease investment cost, decouple services from the underlying information technology and provides elasticity in terms of resource provisioning (Chen, G., D. Kotz, 2000). Security is the main element in any cloud infrastructure, because it is essential to make sure that only authorized and authenticated user access is permitted and secure

#### Open Access Journal

Published BY AENSI Publication

© 2016 AENSI Publisher All rights reserved

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

**To Cite This Article:** S. Srinivasan and K. Raja., Ensuring Cloud Security and Privacy Through Enhanced Merkle Hash Tree Method Using Bat Mechanism. *Aust. J. Basic & Appl. Sci.*, 10(1): 676-684, 2016

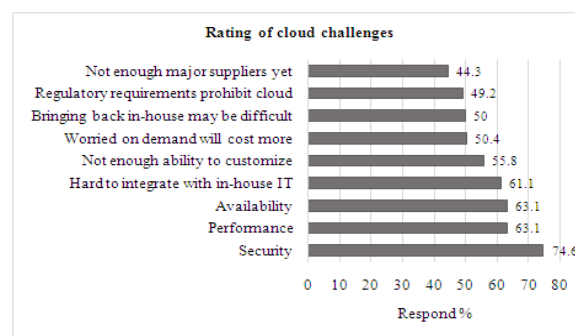
behavior is recognized (Chen, D., H. Zhao, 2012). Eventhough the security and privacy is still uncertainty while sharing of resources and services in cloud computing environment. Some security issues in cloud are dynamic data integrity, information confidentiality, public auditability, loss of data, vulnerability and data intrusion. To ensure facts information confidentially, data integrity and service availability, the cloud service provider provides that at a minimum, include :

- Third party auditing method to guarantee the public auditability and secured all information.
- Stiff user access methods and user authentication technique to protect the confidential data from malicious users and intruders.

Cloud computing security is a massive set of security controls, policies, methods and technologies set to preserve privacy, protect the confidential data and applications of cloud computing. The rest of this paper is organised as follows: The Section II discusses cloud security issues and challenges. Section III, gives a detailed description of the proposed exciting enhanced merkle hash tree method using bat mechanism for cloud computing. Section IV shows performance of experimental results. Finally, Section V concludes the paper s with future work directions.

## II. Security issues and challenges:

Protecting the cloud data such as sharing of users data,resources, user identification like credit card details from the malicious users is a primary impact in cloud. The security (Zhou, M., 2010) is the majority responding in high percentage of challenge of nine issues known to cloud environment as shown in Fig 1.



**Fig. 1:** Challenges of cloud environment.

**Table 1:** Risk distribution with respect to cloud.

Areas of Risk	Critical	Important	Not so important
Data and information security	91.7%	8.3%	0.0%
Change control management	41.7%	50.0%	8.3%
Third party authentication management	41.7%	41.7%	16.7%
Service level Agreement, regulations and legislation	33.3%	41.7%	25.0%
Disaster recovery	66.7%	33.3%	0.0%

Minqi Zhou *et al* [6] have suggested a number of cloud computing system providers about their concerns on security and privacy issues. For security, they have found out these concerns were not sufficient and more should be included in terms of several aspects like service availability, data confidentiality, data integrity, audit and control etc. Moreover, open-minded acts on privacy were out of date to safe guard users information in the cloud computing system environment. As they were no longer relevant to the fresh relationship among users and providers, which includes different parties like cloud user, cloud service provider etc. Multi-located data storage space and services in the cloud build privacy issues even inferior. Therefore, implementing released acts for new circumstances in the cloud, it will effect in more users to step into cloud. They have kept up that the riches in Cloud Computing literature was to be coming after these security and privacy issues having been found out. Fig 1. Challenges of cloud environment. A cloud storage system has been suggested by Hsiao-Ying Lin and Wen-Guey Tzeng (2012) which enclosed storage and key servers. They combine a recently suggested threshold proxy re-encryption scheme and erasure codes over exponents. The threshold proxy re-encryption scheme maintains encoding, forwarding, and fractional decryption operations in an allocated way. In Cloud Computing, a privacy-preserving public auditing system for information storage security has been suggested by Cong Wang *et al* (2013). They employ the homomorphic linear authenticator and random masking to guarantee that the third party auditor would not know any knowledge about the data content accumulated on the cloud server during the capable auditing process, which not only destroys the burden of cloud user from the tedious and probably costly auditing task, however furthermore improves the users' fear of their outsourced data leakage. Arjun Kumar *et al* (Kumar, A., 2012) proposed the secure data storage and data access in cloud. They identified the cloud owner

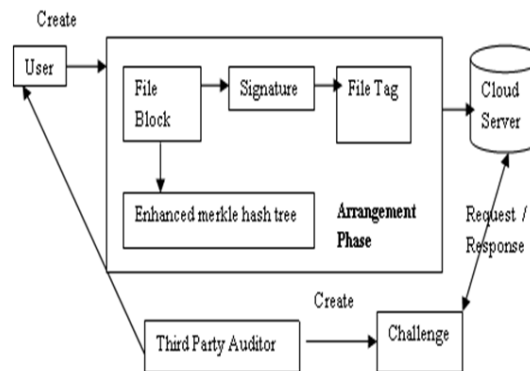
will not have authority to control the data due to lack of user access privilege, third party auditing and non-availability of service.

The important problem on cloud is data integrity. The confidential information stored in the cloud storage may suffer from harm or damage during transition actions from or to the cloud service provider such as the recently assaulted linux's servers (Cachin, C., 2009; Carroll, M., 2011). There are several areas of risks could be identified, in which data and information security was the rate by 91.7% (Denz, R., S. Taylor, 2013) as exposed in Table 1. Ching-Nung Yang and Jia-Bin Lai suggested a cloud security services together with cryptographic key agreement and authentication. They utilized the secure cloud computing through make use of Elliptic Curve Diffie-Hellman and Symmetric Bivariate Polynomial based Secret Sharing mechanism (Nabeel, M., E. Bertino, 2014). Mohamed Nabeel *et al* (2014) proposed privacy preserving access control in public clouds use the ACP method for encryption and upload the data in the remote storage in this method was increase the computational cost. Some of the major cloud computing issues and different attacks (Yang, C.N., J.B. Lai, 2013) are mentioned below:

- Information confidentiality
- Vulnerability
- Public auditability
- Data Leakage and loss of control
- Data intrusion
- Service and Data availability
- Injection and Hypervisor attack

### III. Enhanced merkle hash tree method using BAT mechanism:

The enhanced merkle hash tree method using bat method for cloud computing assesses the problem of security and privacy concerns includes public auditability, dynamic data integrity, data authenticity from the cloud architecture perspective, cloud delivery model and cloud deployment method view point. This method assess the problem of various attacks, data leakage, information privacy, confidentiality and vulnerability while sharing of services and resources in cloud computing environment. This method allows authenticated and authorized users' to access the confidential data, which leads to develop well-organized and proficient cloud security environment. Cloud computing presents appropriate on-demand network access to an allocated pond of computing resources. This restricted method carries several security confronts, which have not been fine implied. The main difficulty in cloud computing is making sure the integrity of data storage. The general architecture for secured data authentication through enhanced merkle hash tree method for cloud computing environment is shown in Fig 2.



**Fig. 2:** Enhanced merkle hash tree general architecture.

This proficient approach is primarily focus on the authentication and auditing of cloud environment. At first, the client generates the initial setup and stores it in the cloud server. To check the public auditability, third party auditor request to the server, then the server response to the request and generate some auxiliary information. This information is send to the client for the further process through arrangement phase of enhanced merkle hash tree method. The arrangement phase create signature for each file block and allotted tag for auditing phase and simultaneously create enhanced hash tree for respective file block. The enhanced merkle hash tree is probable to efficiently and securely show that a set of elements are undamaged and unaltered. For every authentic value of data, a tree will be erected with hashes  $h(\cdot)$  and the tree applied in enhanced merkle hash tree will be a binary tree. The example of enhanced merkle hash tree is shown in Fig 3.

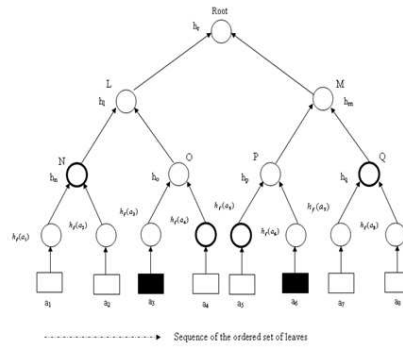


Fig. 3: Example of enhanced merkle hash tree.

This merkle hash tree is normally employed to authenticate the confirmation of data blocks. It utilized one way hash function  $h_f$  to the whole public file instead of signing each entry in the public file. The output of  $h_f$  is called root ( $R_t$ ) of the public file. All the users of the system make out  $R_t$  in hash tree. After that all the users can confirm the accuracy of the public file by computing  $R_t = h(\text{public file})$ . By applying enhanced merkle hash tree, validate individual entries in the public file without having to make out the whole public file. The way of constructing enhanced merkle hash tree is depicted in Fig 3, during the construction of this tree, verifier with the authentic  $h_r$  requests for  $\{a_3, a_6\}$ . The server presents the verifier with the auxiliary authentication information ( $\Omega_i$ ) such as  $\Omega_3 = \langle h_f(a_4) h_n \rangle$  and  $\Omega_6 = \langle h_f(a_5) h_q \rangle$ . Next the verifier can authenticate the  $a_3$  and  $a_6$  by computing  $h_f(a_3)$ ,  $h_f(a_6)$ ,  $h_o = h(h_f(a_3) || h_f(a_4))$ ,  $h_p = h(h_f(a_5) || h_f(a_6))$ ,  $h_l = h(h_n || h_o)$ ,  $h_m = h(h_p || h_q)$  and  $h_r = h(h_l || h_m)$  then making sure if the computed  $h_r$  is similar as the validate one. The bat mechanism is a meta-heuristic algorithm, enthused by the echolocation behavior of micro-bats (Van Slyke, R.M., R. Wets, 1969). The procedure of bat mechanism is used to optimize the encrypted values while constructing the enhanced merkle hash tree as shown in Fig 4.

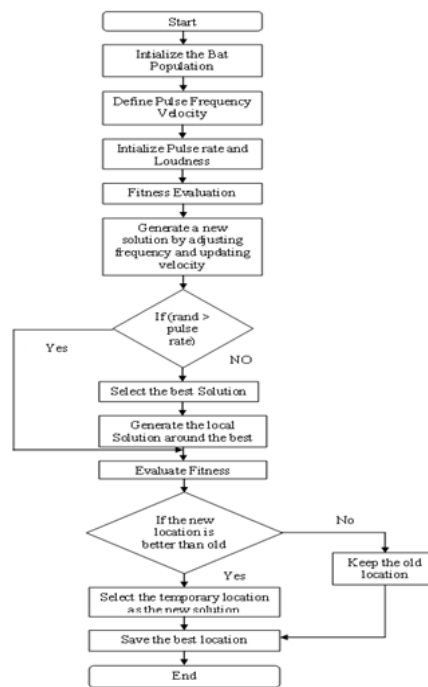


Fig. 4: Procedure of bat mechanism.

The most important goal of this paper as dynamic data integrity and public auditability is mentioned. Initially consider the file  $X$ , next the file is divided into  $m$  number of blocks as  $a_1, a_2, \dots, a_m$  where  $a_i \in Z_p$  and  $p$  is a large prime number. Let  $e : G \times G \rightarrow G_T$  be a bilinear map, with a hash function  $h_f : \{0,1\}^* \rightarrow G$ . Let  $g$  be the generator of  $G$ .  $h$  is a cryptographic hash function. The specified process of this method is carried

out through arrangement phase, integrity phase and dynamic data integrity phase as mentioned below. Arrangement phase. The client make public key and private key (pk, sk) by invoking KeyGen(.). In KeyGen(.), the client choose a random element  $b \leftarrow Z_p$  and calculates  $u \leftarrow g^b$ . The private key pair is (b, sk) and the public key pair is (u, pk). SigGen(.) is run by the client. It takes the input private key sk and file X which is an arranged collection of blocks  $\{a_i\}$ . This arrangement phase has five steps, which can be explained in below:

- Step 1: Generate the tag for file
- Step 2: Create the signature for each block
- Step 3: Generate the signature set
- Step 4: Root of the IMHT is signed using secret key
- Step 5: All the setup send to the server

In a specified file  $X=(a_1, a_2, \dots, a_m)$  the client select a random element  $s \leftarrow G$  and select the file name then choose the tag ( $t_g$ ) for the file X. Let  $t_g = \text{file name} \parallel m \parallel s \parallel \text{SSig}_{pk}(\text{file name} \parallel m \parallel s)$  be the file tag for X next the client calculate signature  $\delta_i$  for each block  $a_i$  ( $i=1, 2, \dots, m$ ) as  $\delta_i = (h_f(a_i) S^{a_i})^b$  then the set of signature indicated as  $\psi = \{\delta_i\}$ ,  $1 \leq i \leq m$ . after client create a root  $R_t$  based on the construction of merkle hash tree, where the leave nodes of the tree are an ordered set of file tag  $h_f(a_i)$  ( $i=1, 2, \dots, m$ ). Next client signs the root  $R_t$  under the private key b:  $\text{SSig}_{sk}(h(R_t)) \leftarrow (h(R_t))^b$  after finishing all the process client send  $\{X, t_g, \psi, \text{SSig}_{sk}(h(R_t))\}$  to the server and delete  $\{X, \psi, \text{SSig}_{sk}(h(R_t))\}$  from its local storage space. After that the client distributes the public key to the third party auditor for monitoring the remote files.

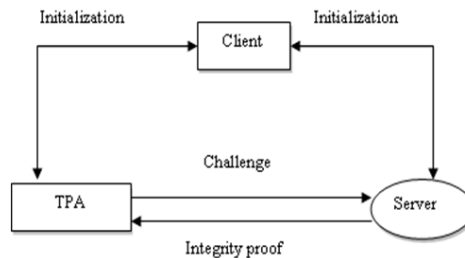


Fig. 5: Representation of data integrity phase.

Integrity phase. By challenging the server, the client or Third Party Auditor (TPA) can validate the integrity of data as shown in Fig 5. The TPA originally verifies the tag through the public key before challenging the server. If the verification fails discard by false; or else recover file name and s. To validate the integrity TPA send the “chal{(i, u<sub>i</sub>)<sub>d<sub>1</sub> < i < d<sub>c</sub></sub>” request to the server. The message “chal” exact the position of the blocks to be verified in the challenge area. To produce the message chal, the third party auditor picks the random c-element subset  $U = \{d_1, d_2, \dots, d_c\}$  of set  $[1, m]$ , where we assume  $d_1 \leq \dots \leq d_c$  for each  $i \in U$  next the TPA select a random element  $U_i \leftarrow B \subseteq Z_p$ . The server make the GenProof(.) after receiving the “chal” request from TPA, and as well compute  $\sigma, \gamma$  (Eq.1 and 2). GenProof(.) taken a input file X, its signature set  $\psi$  and a challenge “chal{(i, u<sub>i</sub>)<sub>d<sub>1</sub> < i < d<sub>c</sub></sub>”

$$\gamma = \sum_{i=d_1}^{d_c} u_i a_i \in Z_p \tag{1}$$

$$\sigma = \prod_{i=d_1}^{d_c} \delta_i^{u_i} \in G \tag{2}$$

Besides the server will also offer the verifier with a small amount of auxiliary information ( $\Omega_i$ )  $d_1 \leq i \leq d_c$  which is the node siblings on the path from the leaves  $\{h_f(a_i)\}$   $d_1 \leq i \leq d_c$  to the root  $R_t$  of the improved merkle hash tree. Next the server responses the verifier with the integrity proof  $I_p = \{ \gamma, \sigma \{ h_f(a_i), \Omega_i \}_{d_1 \leq i \leq d_c}, \text{Sig}_{sk}(h_f(r_i)) \}$ . The verifier produces root  $R_t$  by means of  $\{h_f(a_i)\}_{d_1 \leq i \leq d_c}$  upon receiving the responses from the server, and verification it by checking  $e(\text{Sig}_{sk}(h_f(r_i)), g) = e(h_f(r_i), g^b)$  if the authentication fails, the verifier discard by FALSE; otherwise, the verifier checks (Eq.3)

$$e(\sigma, g) = e(\prod_{i=d_1}^{d_c} h_f(a_i)^{u_i}, s^\gamma, u) \tag{3}$$

if so the output true otherwise false. The overall process of integrity phase is given below,

- Step 1: Random set generation
- Step 2: Challenging the server
- Step 3: Compute delta and  $\sigma$
- Step 4: Server respond the request

Step 5: Calculate  $R_t$  and Verify the Signature

Step 6: Verify  $\{a_i\}_{i \in t_0 \cup}$

Dynamic data integrity phase. This approach can adeptly handle dynamic data integrity together with data modification ( $D_m$ ), data insertion ( $D_i$ ) and data deletion ( $D_d$ ) of cloud storage. Now we consider the file  $X$ , signature  $\psi$  and as well the root  $R_t$  has been signed by the client and all the information collected at the cloud server. So anyone who has the client's public key can look the accuracy of data storage. The data modification refers to the substitute of specific blocks with novel one. If suppose, the client wants to alter the  $i$ th block of  $a_i$  to  $a_i'$  after that the client produces the related signature of the block  $\delta_i = h_f(a_i)$   $S_{a_i}$  then client produce a revise request message  $update = (D_m, i, a_i', s_i)$  and send to the server, where  $D_m$  represents the data modification operation as shown in Fig 6.

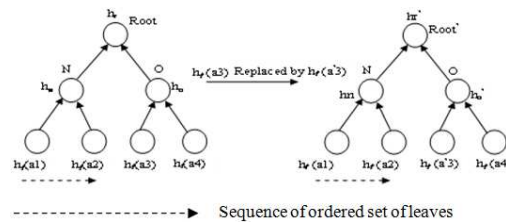


Fig. 6: Enhanced merkle hash tree update under block modification.

The server runs  $ExecUpdate(X, \psi, update)$  upon receiving the appeal. Particularly the server,

- 1) Replace the block of  $a_i$  with  $a_i'$  and output  $X'$ .
  - 2) Replaces  $\delta_i$  with  $\delta_i'$  and output  $\psi'$ .
  - 3) Replaces  $h_f(a_i)$  with  $h_f(a_i')$  in the better merkle hash tree construction and generate the root  $R_t'$ .
- Inserting a novel block  $a^*$  after the  $i$ th block  $a_i$  is known to be data insertion as shown in Fig 7.

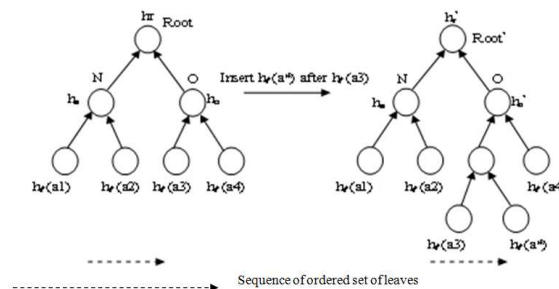


Fig. 7: Enhanced merkle hash tree block insertion

The procedures of data insertion operation are the same as the data modification case. The client produce the related signature  $\delta^*$  then produce an update message  $update = (D_i, i, a^*, \delta^*)$  and sends to the server, where  $D_i$  signify the data insertion operation. The server runs  $ExecUpdate(X, \psi, update)$  upon receiving the appeal. Particularly the server,

- 1) The server stores  $a^*$  and adds a leaf  $h_f(a^*)$  and the output  $X'$ .
- 2) Then generate new root  $R_t'$ .
- 3) Adds the signature  $\delta^*$  into the signature set and output  $\psi'$ .

Data deletion is the converse operation of data insertion. For single block deletion, it refers to deleting the particular block and moving all the final blocks one block forward. If the server obtains the update request for deleting block  $a_i$  from its storage, remove the leaf node  $h_f(a_i)$  in the enhanced merkle hash tree and produce the new root  $R_t'$  as shown in Fig 8.

**IV. Performance And Evaluation:**

This paper carried out the secured authentication by enhanced merkle hash tree is implemented in JAVA technology. The performance of this enhanced merkle hash tree method using bat mechanism is evaluated based on diverse number of sampled blocks through auditing and compare with other methods like HASSM, PPV in terms of various measures includes audit time, bandwidth utilization, computational cost, user queried rate etc. This method verifies data integrity, authentication of resources which uses binary tree structure and the third

party auditor verifies public auditability with multiple auditing task simultaneously. The Table 2 and Fig 9 represents of the Audit time per task for proposed method attains better value when compared to other methods.

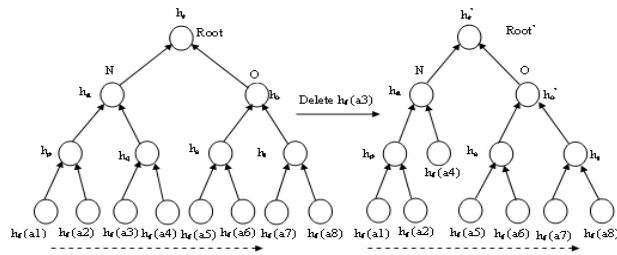


Fig. 8: Enhanced merkle hash tree update under block deletion.

Table 2: Shows tabulation of Audit Time per Task.

No. of audit task	Audit Time per Task (ms)		
	Proposed method	PPV	HASSM
6	389	520	400

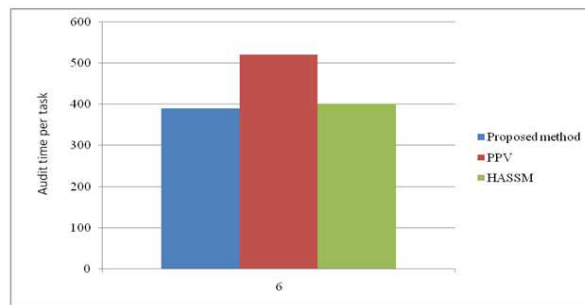


Fig. 9: Comparison of Audit time per task.

The Table 3 and Fig 10 shows the values of bandwidth utilization for the corresponding block size. Bandwidth utilization is defined as the bit rate variable consumed to perform the data communication in the cloud infrastructure. The bandwidth utilization of the proposed method is more efficient than other methods.

Table 3: Bandwidth utilization of various methods.

Block Size (KB)	Bandwidth Utilization (M bits / s)		
	Proposed method	PPV	HASSM
100	27651	23150	25000

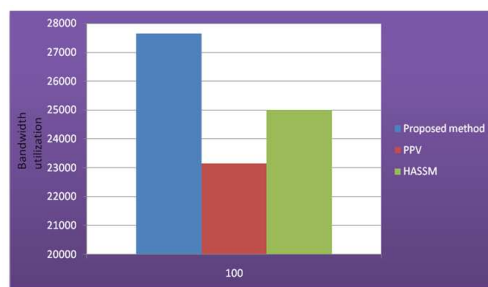
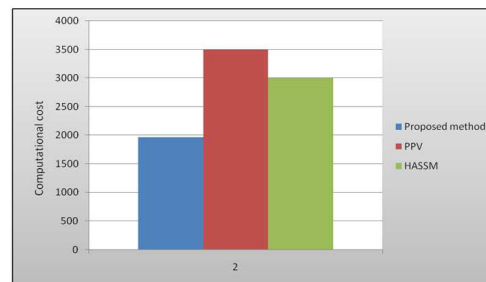


Fig. 10: Bandwidth utilization for proposed and other methods.

The Table 4 represents the values of computational cost for the various file size. 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). The computational cost of the proposed method is compared with various existing method as shown Fig 11.

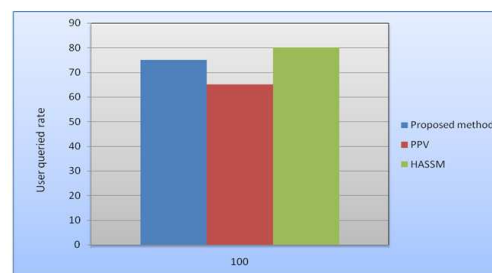
Table 4: Reduction of computational costs.

File Size (KB)	Computational Costs (KB)		
	Proposed method	PPV	HASSM
2	1964	3495	2998



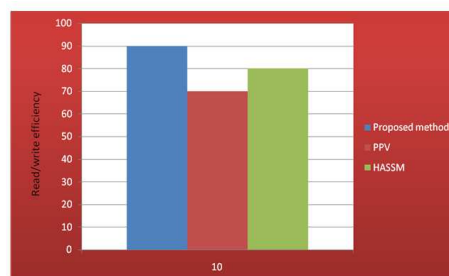
**Fig. 11:** Graphical representation of computational costs.

The user queried rate denotes the amount of effectiveness in producing the result to the end users. The user queried rate of the proposed method is more efficient than other methods as shown in Fig 12.



**Fig. 12:** Graphical representation of User queried rate.

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 on Cloud Services of the proposed method is compared with other methods as shown in Fig 13.



**Fig. 13:** Read / Write efficiency of various methods.

In order to decrease several risk in a cloud environment of any organization, the proposed cloud secure method use better, highly safe to protect the data, sharing of computing services and resources in cloud computing environment.

#### **V. Conclusion and future work:**

Clearly, present growth of cloud environment has tremendously increased day-by-day, but the cloud security and privacy are still measured and it has been important key concern in the cloud computing. To protect the confidential data in cloud against verification and validation of users, threats, vulnerabilities, therefore a secure dynamic authentic security for cloud environment is enforced. The efficient dynamic data integrity and secure authentication can be achieved through enhanced merkle hash tree construction using bat algorithm. To solve the public auditability, the proposed method use third party auditor for multiple auditing tasks simultaneously. The experimental result shows that the proposed method is highly proficient and provably secure. This progressive research work enriched efficiency, well bandwidth utilization, improved auditing time and frequency, decrease of computational cost by implementing enhanced merkle hash tree using bat mechanism in cloud computing. Future research work on this work to develop better cryptographic mechanism

with multicloud storage method that can support high confidentiality, data availability and to meet more privacy and secure cloud computing environment.

## REFERENCES

- Lin, G., 2012. Research on electronic data security strategy based on cloud computing. In *Consumer Electronics, Communications and Networks (CECNet), 2nd International Conference on* (pp: 1228-1231). IEEE.
- Mohamed, E.M., H.S. Abdelkader, S. El-Etriby, 2012. Enhanced data security model for cloud computing. In *Informatics and Systems (INFOS), 8th International Conference on* (pp: CC-12). IEEE.
- Behl, A., K. Behl, 2012. An analysis of cloud computing security issues. In *Information and Communication Technologies (WICT), 2012 World Congress on* (pp: 109-114). IEEE.
- Chen, G., D. Kotz, 2000. *A survey of context-aware mobile computing research*, 1(2.1): 2-1. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.
- Chen, D., H. Zhao, 2012. Data security and privacy protection issues in cloud computing. In *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, 1: 647-651. IEEE.
- Zhou, M., R. Zhang, W. Xie, W. Qian, A. Zhou, 2010. Security and privacy in cloud computing: A survey. In *Semantics Knowledge and Grid (SKG), 2010 Sixth International Conference on* (pp: 105-112). IEEE.
- Lin, H.Y., W.G. Tzeng, 2012. A secure erasure code-based cloud storage system with secure data forwarding. *Parallel and Distributed Systems, IEEE Transactions on*, 23(6): 995-1003.
- Wang, C., S.S. Chow, Q. Wang, K. Ren, W. Lou, 2013. Privacy-preserving public auditing for secure cloud storage. *Computers, IEEE Transactions on*, 62(2): 362-375.
- Kumar, A., B.G. Lee, H. Lee, A. Kumari, 2012. Secure storage and access of data in cloud computing. In *ICT Convergence (ICTC), 2012 International Conference on* (pp: 336-339). IEEE.
- Cachin, C., Keidar, I., & Shraer, A. (2009). Trusting the cloud. *Acm Sigact News*, 40(2): 81-86.
- Carroll, M., A. Van Der Merwe, P. Kotze, 2011. Secure cloud computing: Benefits, risks and controls. In *Information Security South Africa (ISSA)*, (pp: 1-9). IEEE.
- Denz, R., S. Taylor, 2013. A survey on securing the virtual cloud. *Journal of Cloud Computing*, 2(1): 1-9.
- Yang, C.N., J.B. Lai, 2013. Protecting data privacy and security for cloud computing based on secret sharing. In *Biometrics and Security Technologies (ISBAST), 2013 International Symposium on* (pp: 259-266). IEEE.
- Nabeel, M., E. Bertino, 2014. Privacy Preserving Delegated access control in Public clouds. *Knowledge and Data Engineering, IEEE Transactions on*, 26(9): 2268-2280.
- Govindaraj, D.T., E. Viswanathan, 2014. Bat optimization algorithm for security constrained optimal power flow. *International Journal of Advanced and Innovative Research*, 3(1).
- Van Slyke, R.M., R. Wets, 1969. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics*, 17(4): 638-663.