European Journal of Scientific Research
ISSN 1450-216X / 1450-202X Vol. 147 No 3 November, 2017, pp. 275-286
http://www. europeanjournalofscientificresearch.com

Integrity Rule Generat ion and Tiger Hashing Technique for Efficient and Secure Cloud Data Storage

M. Buvaneswari
Research Scholar, Department of Computer Science
Bharathiar University, Coimbatore,Tamilnadu, India
E-mail: buvanamscmphil@gmail.com

N. Rajendran
Principal, Vivekanandha Arts and Science College for Women
Sankari, Tamilnadu, India
E-mail: vpnraj@gmail.com

Abstract

Security in cloud data storage is a significant when using the cloud services provided by the service provider in cloud environment. Most of research works has been developed for cloud data storage. But, cloud users still have security and privacy problems about their outsourced data due to potential unauthorized access. In order to overcome such limitation, Tiger Hashing Integrity Rule (THIR) model is proposed. The THIR model is designed for secure cloud data storage. Initially, the request is sent from the cloud users to cloud server. Next, Tiger Hashing technique is used for constructing the Tiger Hash Tree to achieve efficient data storage and data dynamics for improving the cloud data storage capacity and minimizing the user task service rendering complexity. Finally, Tiger Hashing Signature is used to verify the integrity of the data from the cloud server. This helps to improve the security and data integrity of cloud storage services in an effective manner. The performance of THIR model is measured in terms of data storage capacity, cloud data security, accessing time and data integrity rate. The experimental result demonstrates that the THIR model is able to improve the cloud data security and also improves the data integrity rate when compared to state-of-the-art-works.

Keywords: cloud data storage, cloud users, tiger hash tree, tiger hashing signature, data integrity, security

I. Introduction

Cloud computing offers a low-price, scalable, location free infrastructure for data management and storage. Besides, cloud storage is a growing popular application of cloud computing that offer on-demand outsourcing data services for both organizations and individuals. The key problem in cloud data storage is security due to potential unauthorized access within cloud service providers. Therefore, there is a need for new model to achieve efficient and secure cloud data storage in cloud computing. Ease of Use

Recently, many research works have been planned for secure cloud data storage and data integrity verification. For example, an efficient and secure public verification of data integrity scheme was designed in [1] for guarding outsourced data from the external adversaries and malicious auditors which resulting in improved data integrity rate. However, security of outsourced data was not

sufficient. A novel public auditing scheme for secure cloud storage using dynamic hash table (DHT) called Dynamic Hash Table- Public Auditing (DHT-P) was developed in [2] that used new two dimensional data structure to record the data property information for dynamic auditing and to reduce the costs of storage. But, the accessing time was higher.

Public auditing protocol was designed in [3] to present secure cloud storage service to users and to authenticate data integrity. The Public auditing protocol improves the security. However, the public auditing model for cloud storage and the model against the pollution attacks for linear network coding were remained unaddressed. A Dual-Server Public key Encryption with Keyword Search (DSPEKS) was developed in [4] to solve the security vulnerability in cloud data storage. But, the security and efficiency of DSPEKS was insufficient.

A Cipher text Policy Attribute-Based Encryption (CP-ABE) based data sharing scheme was presented in [5] to deal the security problem in cloud environment and to achieve secure cloud storage. However, the performance of encryption and decryption process is not efficient. A probabilistic challenge–response scheme was intended in [6] to afford an efficient way to identify malicious behavior of cloud servers and to improve the security and reliability of cloud storage. Although, computation and communication overhead was not at required level.

An effective and flexible distribution verification mechanism was planned in [7] to attain higher data storage security in cloud computing and to verify the correctness of users 'data in cloud data storage. A Cryptographic Role-Based Access Control scheme was designed in [8] to afford scalable high-performance storage architecture and to lessen the cost of maintenance of individual services. A secure data integrity checking decentralized erasure code cloud storage system based on the threshold share scheme was introduced in [9] for defending cloud storage servers from modifying data.

A secure cloud storage system was presented in [10] to improve the authentication level of security with the aid of two authentication techniques namely time-based one-time password (TOTP) for cloud users authentication and automatic blocker protocol (ABP) to defend the system from unauthorized third party auditor.

In order to overcome the above mentioned existing issues, a novel model called Tiger Hashing Integrity Rule (THIR) model is developed. The key objective of THIR model is to enhance cloud data capacity with minimal data accessing time.

The contribution of the research work is formulated as

- To improve the cloud data storage capacity and security with fast accessing of data, a tiger hash tree structure is designed in THIR model.
- To verify the accuracy of cloud users data obtained from the cloud server, tiger hashing signature is employed in THIR model.

The rest of the paper organized as follows. In Section 2, the proposed THIR model is explained with the help of neat architecture diagram. In Section 3, the experimental setting is discussed with exhaustive analysis of results described in Section 4. In Section 5, a review of different techniques designed for cloud data storage and data integrity checking is discussed with their limitations. In Section 6, the concluding remark is presented.

II. Tiger Hashing Integrity Rule Model for Secure Cloud Data Storage

The Tiger Hashing Integrity Rule (THIR) model is designed to improve the security of cloud data storage and to improve the data integrity in cloud computing. The THIR model is used Tiger Hashing Technique for efficient cloud data storage. The cloud data storage architecture consisting of three entities namely cloud user, cloud service provider, third party auditor as shown in below Figure 1.

Figure 1: Architecture Diagram of Green Cloud Data Storage

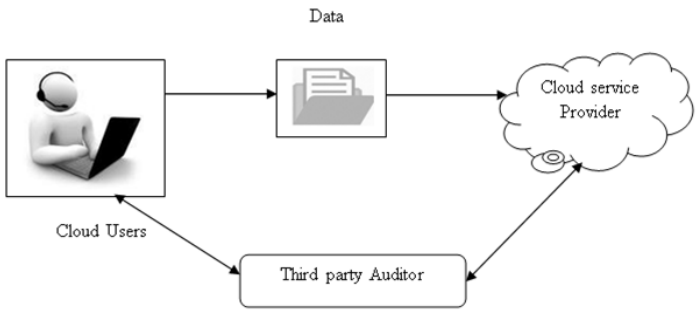
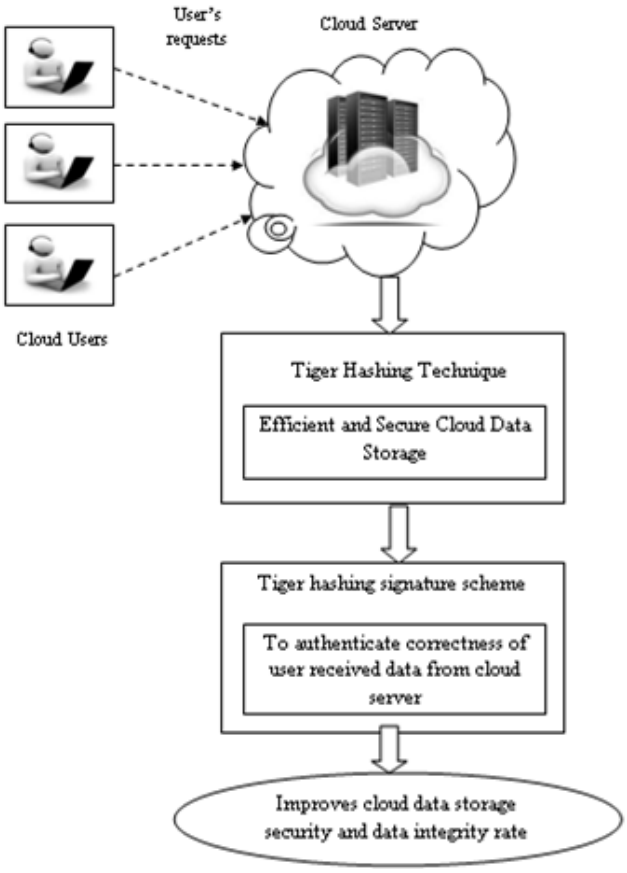


Figure 1 shows the architecture diagram of could data storage in which the cloud service provider comprises the collection of software and hardware resources and provides flexible online computing and data storing services. The cloud users outsource their data to the remote cloud storage for saving the storing cost and utilize the computation resources offered by the service provider. The third- party auditor is a partially trusted and independent entity that can provide assess and arbitration if essential. The cloud users interact with cloud server for accessing and updating data stored in cloud.

To improve the efficiency and security of cloud data storage, THIR model is developed. The main objective of THIR model is to achieve higher secure cloud data storage and thereby improving the speed of data accessibility by using Tiger Hashing Technique. The architecture diagram of THIR model is shown in below Figure 2.

Figure 2: Architecture Diagram of THIR Model for Secure Cloud Data Storage



As shown in Figure 2, initially the requests are sent from the cloud users to cloud server. Next, THIR Model is employed tiger hashing technique for storing the user requested data. Finally, Tiger

hashing signature scheme is employed to verify the integrity of user received data from the cloud server. This in turn helps for improving the security of cloud data storage and integrity in an efficient manner.

A. Tiger Hashing Technique

The THIR model is used Tiger Hashing Technique for efficient cloud data storage. Tiger Hashing Technique is a Tiger hash tree structure in which each non-leaf node includes the hash value of its children nodes and each leaf node consisting of hash value of a data block. There is a root on the top of the tiger hash tree. The purpose of using tiger hash tree in THIR model is to improve cloud data storage capacity with minimal data accessing time and to attain higher security of data storage in cloud. The tiger hash tree is employed for secure cloud data storage. The tiger hash tree is efficient since they employ hashes instead of the data files. The tiger hash tree protects the integrity of cloud data and support dynamic maintenance. In tiger hash tree, every node keeps the location of the corresponding node, so that the cloud users can authenticate the consistency of the challenge response blocks through computing the root value directly without retrieving the entire tree. A general tiger hash tree structure that includes n leaf nodes is mathematically expressed as,

$$THT = \{Node_i \mid NodeN_i\} = h(data_i),\ 1 \le I \le n \quad (1)$$

From (1), $Node_i$ indicates the number of nodes in tree and h () represents a hash function where $data_i$ designates the different data stored in tiger hash tree. Afterward, the value of the non-leaf node $Node_i$ is formulated as

$$Node_i = h\left(Node_l^i | Node_r^i\right) \quad (2)$$

From (2), $Node_l{}^i$ and $Node_i{}^r$ indicates left child and right child nodes respectively. The root node of tiger hash tree is represented as N_{root}. Given a node $Node_i$, the smallest ordered node set $\Omega_{i=}$ { $Node_1{}^i >> Node_2{}^i >> \ldots..$ } which can used by $Node_i$ o compute the root node N_{root} is called auxiliary authentication information (AAI). A tiger hash tree structure for efficient cloud data storage is demonstrated in below Figure 3.

Figure 3: Tiger Hash Tree for Cloud Data Storage

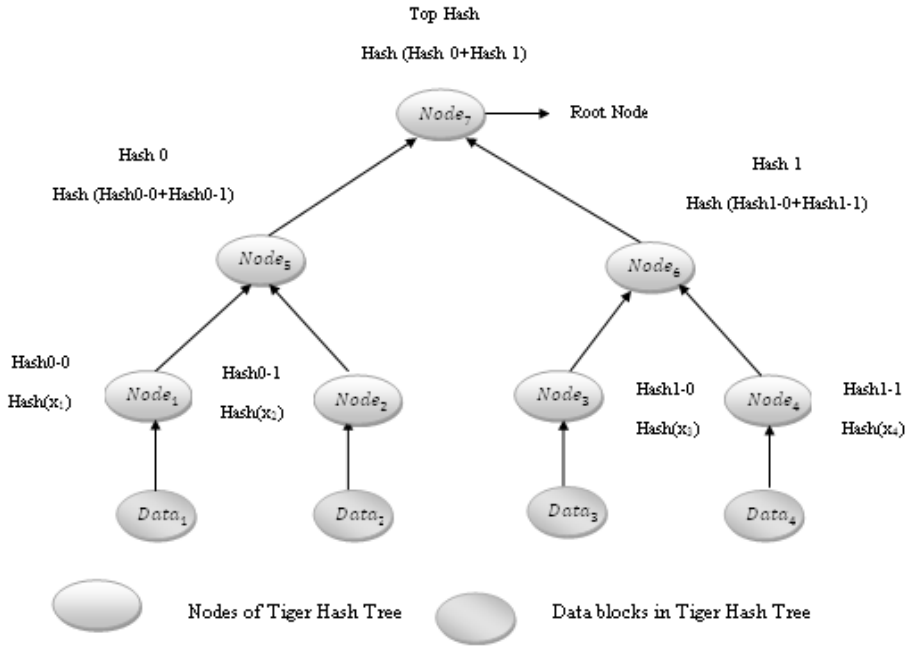


Figure 3 shows the tiger hash tree structure for cloud data storage whereas every non-leaf nodes are labeled with the hash value of its children nodes and all leaf nodes are labeled with the hash value

of a data block. Therefore, the THIR model improves the data storage capacity and security of the data in cloud computing. The tiger hash tree structure permits the cloud users to perform the dynamic operations such as insert, delete etc. The algorithmic process of tiger hash tree for efficient cloud data storage is revealed in below,

Algorithm 1: Tiger Hashing Tree Based Cloud Data Storage

```
//Tiger Hashing Tree Based Cloud Data Storage Algorithm
Input: Nodes  Nodei = Node1 , Node2 , Node3, Node4 , … .. Noden
Output: Improved cloud data capacity with minimum accessing time
Step 1:Begin
Step 2: create tiger hash tree by using (1)
// Insertion Operation
Step 3: : If the tiger hash tree is not full, then add the data objects
Step 4:     assign new leaf node and then add the newly obtained data objects
Step 5:     insert operation = THT[add new(Node1 , Node2 , Node3, Node4 ,
… .. Noden)]
Step 6:     If the root node divides, then the root node has one key and two
intermediate nodes
Step 7:     Repeat step until no iteration is found
Step 8: End if
// Removal Operation
Step 9: Root node on tiger hash tree where it belongs
Step109: Eliminate the unwanted leaf node as
Removal operation = THT[remove old(Node1 , Node2 , Node3, Node4 , … ..
Noden)]
Step 11: If the leaf node removed, then the leaf node has two intermediate
nodes
Step 12: Repeat step until no iteration is found
Step 13: End
```

With the help of above algorithmic process, THIR model constructs efficient cloud data storage system which allows the cloud users to store the data and work with the data without any complexity, trouble of the resources. As a result, THIR model improves the data storage capacity with minimum data accessing time.

B. Tiger Hashing Signature for Data Integrity Verification

The THIR model is designed to check cloud data integrity based on the tiger hash tree. The tiger hash tree is a binary tree that has the hash values of the data blocks as its leaves. Both the user and cloud server store the data in tiger hash tree. At any time, the cloud user may need to verify his/her data blocks from the server by using tiger hash tree. The THIR model is used tiger hashing signature for verifying the integrity data stored in cloud. The data integrity verification process in THIR model consists of three parts namely Key Generation, Signature Generation and signature verification. The Key Generation algorithm is used to generate the public and secret key for each data file. The Signature Generation is used to generate the signature for each data block. Finally, the signature verification is employed to authenticate the generated signature.

1. Key Generation

The tiger hashing signature can only be utilized to sign a limited number of messages with one public key $\boldsymbol{PK}$. Let us consider $X = X_1, X_2, X_3, \ldots. X_N$ be the set of data files in one folder stored on cloud storage. Let number of files to be power of two, so that we signify possible number of messages as $N = 2n$. For each data files, initially the public key PK_i and secret keys SK_i are generated and then for every public key PK_i with, $1 \leq i \leq 2^n$, a hash value $h_i = H(PK_i)$ is evaluated. With the support of these hash values h_i, a tiger hashing tree is constructed as shown in below Figure 4. The node of tree is termed as

$N_{i,j}$ where i signifies the level of the node. The level of a node is characterized by the distance from the node to a leaf. Therefore, a leaf of the tiger hashing tree has level i = *0* and the root of tiger hashing tree has level $i = n$. In the tiger hashing tree, the hash values h_i are leafs of a binary tree, therefore $h_i = N_{0,i}$. Every inner node of the tiger hashing tree is the hash value of the concatenation of its two children. Hence,

$$N_{0,i} = H(N_{0,0} \| N_{0,1}) \quad (3)$$

In this manner, a tiger hashing tree with 2^n leafs and 2^{n+1} nodes is constructed. The root of the tree $N_{1,0}$is the public key $N_{1,0}$ of the tiger hashing signature scheme.

2. Signature Generation

To sign a data X with the Tiger Hashing Signature, the data X is initially signed with a one-time signature scheme by using the public key PK_i which resulting in a signature $sign'$. The corresponding leaf node (i.e. the node which the user requested data) of the tiger hashing tree is hashing with a one-time secret key $SK_{i\,i}$ which is formulated as

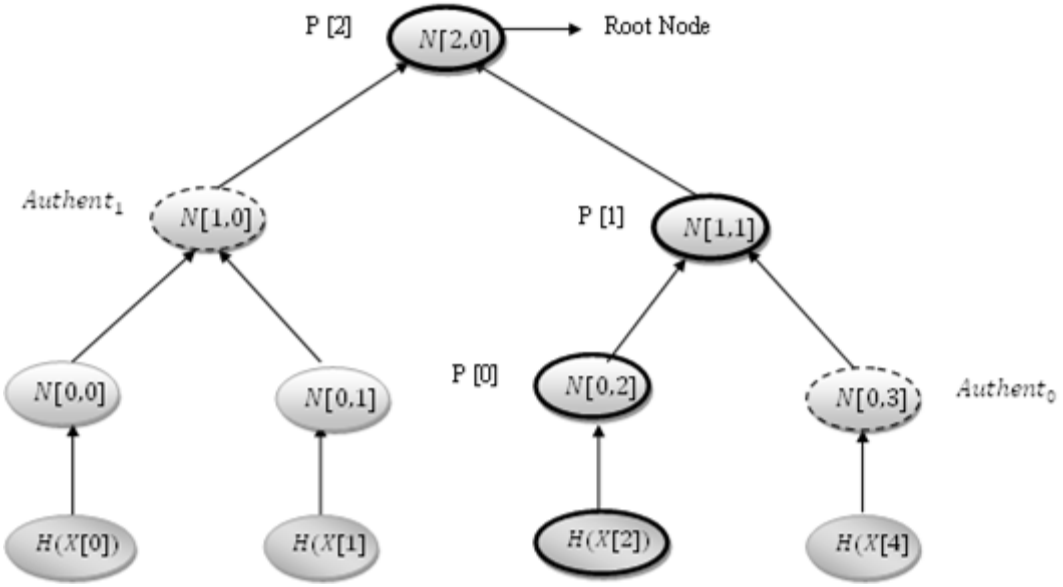
$$N_{0,i} = H(SK_i) \quad (4)$$

Then, call path in the tiger hashing tree from $N_{0,1}$ to the root P. The path P includes $n+1$ nodes with $P_0 = N_{0,i}$ being the leaf and $P_n = N_{n,0} = PK$ being the root of the tree. To estimate this path P all children of the nodes P_0 , P_1 , ……..P_n are needed. We know that P_i is a child of P_{n+1} . To compute the next node P_{i+1} of the path P, both children of P_{i+1} are needed known. Therefore, the brother node of P_i is required and terms this node $Authent_i$, so that $P_{i+1} = H(P_i \| Authent_i)$. Thus, n nodes $Authent_0$,…. $Authent_{n-1}$ are required to compute every node of the path P. The authenticated nodes $Authent_0$,…. $Authent_{n-1}$, secret key SK_i,public part of the key PK_i and the one time signature $sign'$ *are* combined to form the signature by using tiger hashing signature scheme which is formulated as follows,

$$Sign = (Sign' \| PK_i \| SK_i \| Authent_1 \| ….. \| Authent_{n-1}) \quad (5)$$

From the equation (5), the tiger hashing signature is generated and this signature is transmitted to the corresponding users. The tiger hashing tree structure for authentication path is shown in below Figure 4

Figure 4: Tiger Hashing tree with path P and authentication path for i=2



The algorithmic process of tiger hashing signature generation process is shown in below,

Algorithm 2: Tiger Hashing Signature Generation

```
//Tiger Hashing Signature Generation Algorithm
Input: Public KeysPK_i , Secret Key SK_i, data file X = X_I,X_2,X_3, …. X_N
Output: Tiger Hashing signature sign
Step 1: Begin
Step 2:The data X_i is signed with a one-time signature by using public key PK_i for generating sign'
Step 3: The corresponding leaf node of the tiger hashing tree is hashing with the secret key SK_i using (4)
Step 4: Find the authenticated node for generating the tiger hashing signature
Step5: Generate tiger hashing signature using (5)
Step 6:End
```

With the help of above algorithmic process, THIR model generates the tiger hashing signature for data integrity verification.

3. Signature Verification

Signature verification process is performed in receiver side to validate the integrity of received data from the cloud server. The receiver is familiar with the public key PK_i, the data X and the signature. Initially the receiver validates the one time signature of the data with the support of the one-time signature public key PK_i. If the $sign'$ is correct signature of data X, then the receiver evaluates the $P_0 = H(SK_i)$ by hashing the secret key of the one-time signature. For $j = 1,\ldots n-1$, the nodes of P_j of the path are determined with $P_j = H(P_{j-1} \,||\, Authent_{j-1})$. If P_n matches the public key PK_i of the tiger hashing signature, then the signature is valid. The algorithmic process of signature verification is shown in below,

Algorithm 3: Signature Verification Algorithm

```
// Signature Verification Algorithm
Input: Tiger Hashing signature Sign
Output: Improved Data Integrity Rate
Step 1: Begin
Step 2: Receiver authenticates one time signature of data by using the one-time signature public key PK_i
Step 3: If Sign'' is correct signature of data X then
Step 4: Receiver evaluates P_0 = H(SK_i) by hashing the secret key of the one-time signature
Step 5: For j = 1 ,… n – 1, the nodes of P_j of the path, find the authenticated node and generates the tiger hashing
signature New Sign
Step 5:     If New Sign == New Sign
Step 6:         data integrity is achieved
Step 7:     Else
Step 8:         data integrity is not achieved
Step 9:     End if
Step 10:End
```

With the aid of the above process, the THIR model efficiently authenticates the integrity of data stored in cloud which resulting in the improved data integrity rate and achieves higher cloud data security in an effective manner.

III. Experimental Settings

The Tiger Hashing Integrity Rule (THIR) model is implemented in Java Language by using Amazon Access Samples dataset. The CloudSim simulator toolkit has been utilized as a simulation platform with 8 GB of RAM and 1 TB of storage space. Amazon Access Samples dataset information is exploited on the transaction processing between cloud users and cloud servers. The performance of THIR model is compared against with exiting two methods namely Dynamic Hash Table- Public Auditing (DHT-P) [1] and an efficient and secure public verification of data integrity scheme [2]

respectively. The experimental evaluation using THIR model is conducted on different factors such as data storage capacity, cloud data security, accessing time and data integrity rate.

IV. Results and Discussions

The effectiveness of THIR model is compared against with exiting two methods namely Dynamic Hash Table- Public Auditing (DHT-P) [1] and an efficient and secure public verification of data integrity scheme [2] respectively. The performance of THIR model is evaluated along with the following metrics.

A. Measurement of Data Storage Capacity

In THIR model, Cloud data storage capacity refers to the data storage performed in cloud environment based on the different number of user request T, data to be stored (i.e. transaction size) and the time taken for data storage. The cloud loud data storage capacity is measured in terms of kilo bits per second (kbps) and is mathematically formulated as given below.

$$\text{Cloud data storage capacity} = T * d * \text{time} \quad (6)$$

From the equation (6), the cloud data storage capacity is obtained where 'd' represents the data to be stored with respect to time '*time*' respectively. When the cloud data storage capacity is higher, the method is said to be more efficient.

Table 1: Tabulation for Data Storage Capacity

Number of user request	Data Storage Capacity (kbps)		
	DHT-P	**An efficient and secure public verification of data integrity scheme**	**THIR model**
5	66.8	78.5	90.4
10	67.2	79.4	91.2
15	69.3	81.4	92.5
20	71.5	82.9	93.7
25	73.9	83.5	94.2
30	74.8	84.5	95.1
35	76.2	86.4	96.8
40	77.4	88.2	97.3
45	79.7	90.1	98.4
50	81.2	91.8	99.7

Table 1demonstrates the comparative results analysis of data storage capacity using three methods based on the different numbers of user requests in the range of 5-50. While the 20 number of user request is sent to the cloud server, THIR model has achieved 93.7 kbps data storage capacity whereas DHT-P and an efficient and secure public verification of data integrity scheme has achieved 71.5 kbps and 82.9 kbps respectively. Thus, the data storage capacity using THIR model is higher as compared to other exiting methods.

Figure 5: Measurement of Data Storage Capacity

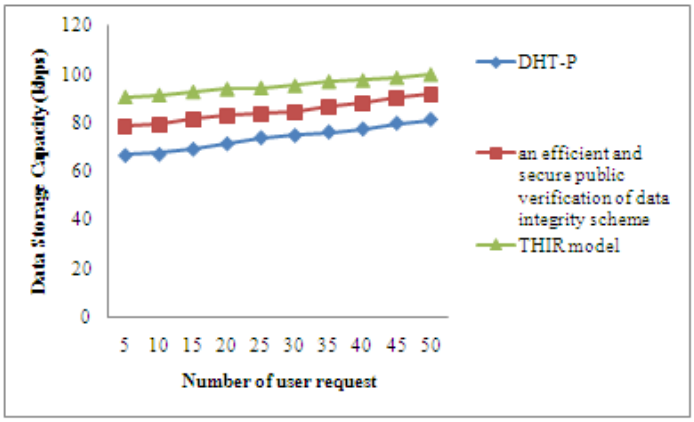


Figure 5 explains the impact of data storage capacity versus diverse number of user requests in the range of 5-50. As exposed in figure, the proposed THIR model provides better data storage capacity as compared to DHT-P [1] and an efficient and secure public verification of data integrity scheme [2]. Besides, while increasing the number of user request, data storage capacity is also increased using all the three methods. But comparatively, data storage capacity using THIR model is higher. This is because of the application of tiger hashing technique in THIR model where the each node stores the hash value of the data instead of storing the data. Furthermore, Security of data stored in tiger hashing tree depends on the security of the hash function. This in turn helps for improving the data storage capacity in an effective manner. Therefore, the THIR model improves the data storage capacity by 22% as compared to DHT-P [1] and 11% as compared to an efficient and secure public verification of data integrity scheme [2] respectively.

B. Measurement of Accessing Time

In THIR model, accessing time computes the amount of time required for acquiring the user request data from the cloud servers in cloud environment. The accessing time is measured in terms of milliseconds (ms) and mathematically represented as follows,

$$\text{accessing } time = n * (time\ for\ acquiring\ single\ user\ request\ data\ ) \quad (7)$$

From the equation (7), the time for gaining the user requested data from the cloud server is evaluated whereas n denotes number of cloud user requests. While the accessing time is lower, the method is said to be more efficient.

Table 2: Tabulation for Accessing Time

Number of user request	Accessing Time (ms)		
	DHT-P	an efficient and secure public verification of data integrity scheme	THIR model
5	15	13	9
10	25	19	15
15	31	26	21
20	39	35	25
25	46	42	29
30	55	51	35
35	69	60	39
40	80	72	45
45	86	80	54
50	95	89	61

The comparative results analysis of data accessing time using three methods based on the diverse numbers of user requests in the range of 5-50 is presented in Table 2. While the 25 number of user request is sent from the cloud users to the cloud server, THIR model has takes 29 ms for accessing the data from the cloud server whereas DHT-P and an efficient and secure public verification of data integrity scheme takes 46 ms and 42 ms respectively. Therefore, the data accessing time using THIR model is lower as compared to other exiting methods.

Figure 6: Measurement of Accessing Time

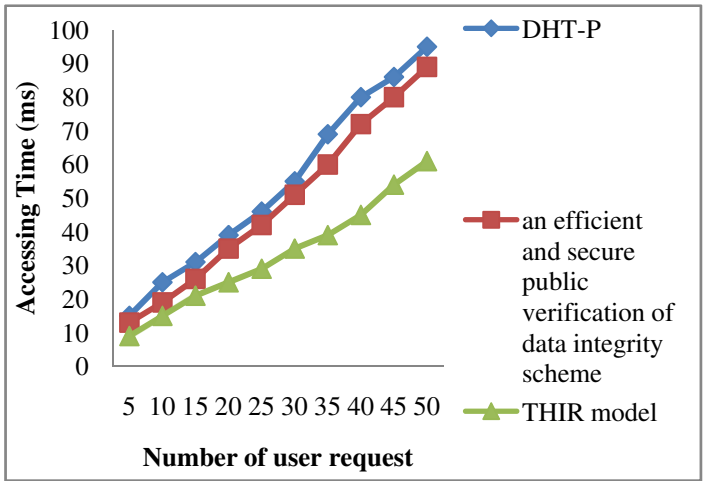


Figure 6 portrays the impact of data accessing time versus different number of user requests in the range of 5-50. As revealed in figure, the proposed THIR model provides better accessing time as compared to DHT-P [1] and an efficient and secure public verification of data integrity scheme [2]. As well, while increasing the number of user request, accessing time is also increased using all the three methods. But comparatively, accessing time using THIR model is lower. This is due to the application of tiger hashing technique in THIR model. Tiger hashing technique employed tiger hash tree structure for effective cloud data storage in which each node keeps its relative position to its parent node and its location information is bound with the value of each node. Therefore, the data are easily extracted from the tiger hash tree with minimum time. This in turn assists for reducing the data accessing time in a significant manner. As a result, the THIR model reduces the accessing time by 62% as compared to DHT-P [1] and 43% as compared to an efficient and secure public verification of data integrity scheme [2] respectively.

V. Related Works

A novel trust model was designed in [11] to improve the security for stored data in cloud storage systems where it employs cryptographic RBAC schemes to verify the reliability of individual roles and users. An enhanced security mechanism based on erasure code was planned in [12] to achieve high availability, strong security in cloud data storage. The problems associated to the cloud data storage for example data breaches, data theft, and unavailability of cloud data and potential solutions to respective issues in cloud was presented in [13].

A Data Partitioning Technique was developed in [14] for offering an efficient data storage security for cloud service and providing the flexible data access. But, the level of security was not efficient. A novel data storage system was intended in [15] to provide a mechanism for securing the data in cloud computing with the assist of Blowfish & RSA algorithms and hash function and to afford

integrity, authentication and confidentiality to the data stored in cloud computing. But, integrity prevention mechanism remained unaddressed.

An efficient and secure dynamic auditing protocol was designed in [16] defends the data privacy against the auditor through combining the cryptography method with the bilinearity property of bilinear paring. Though, the dynamic operations formulate the auditing protocols insecure. A privacy-preserving and auditing-supporting outsourcing data storage scheme was intended in [17] by using encryption and digital watermarking to preserve the privacy of outsourcing data and to protect the data integrity in cloud computing. The data privacy rate is not at required.

A Cooperative Provable Data Possession (CPDP) scheme was presented in [18] by using the homomorphic verifiable response and hash index hierarchy to provide dynamic scalability on multiple storage servers in cloud environment. Though, computational complexity was higher. A data integrity checking algorithm was introduced in [19] to defend static and dynamic data from unauthorized observation, modification, or interference with the aid of hash function. A ciphertext-policy attribute-based encryption (ABE) scheme and a proxy re-encryption scheme were designed in [20] to solve the problems of user revocation and allow the data owner to delegate file re-encryption to cloud servers.

VI. Conclusion

An efficient Tiger Hashing Integrity Rule (THIR) model is designed for secure cloud data storage in cloud computing. At first, the cloud users request is sent to cloud server. Then, Tiger Hash Tree is created by using tiger hashing technique to attain efficient data storage and performing the data dynamics which resulting in increased cloud data storage capacity. At last, the integrity of the data is obtained from the cloud server is authenticated with the support of tiger hashing signature scheme which in turn improves the security and data integrity of cloud storage services in an efficient manner. The efficiency of THIR model is test with the metrics such as data storage capacity, cloud data security, accessing time and data integrity rate. With the experiments conducted for THIR model, it is observed that the cloud data storage capacity provided more accurate results as compared to state-of-the-art works. The experimental results demonstrates that THIR model is provides better performance with an improvement of data storage capacity by 17% and also improves the data integrity rate by 21% when compared to the state-of-the-art works.

References

[1] Hui Tian, Yuxiang Chen, Chin-Chen Chang, Hong Jiang, Yong feng Huang, Yonghong Chen, Jin Liu, “Dynamic-Hash-Table Based Public Auditing for Secure Cloud Storage”, IEEE Transactions on Services Computing, Volume PP, Issue 99, 2016

[2] Yuan Zhang, Chunxiang Xu, Hongwei Li, Xiaohui Liang, “Cryptographic Public Verification of Data Integrity for Cloud Storage Systems”, IEEE Cloud Computing, Volume 3, Issue 5, Pages 44 – 52, 2016

[3] Hongwei Liu, Peng Zhang and Jun Liu, “Public Data Integrity Verification for Secure Cloud Storage”, Journal of Networks, Volume: 8, Issue: 2, Pages: 373-380, February: 2013

[4] Rongmao Chen, Yi Mu; Guomin Yang, Fuchun Guo, Xiaofen Wang, “Dual-Server Public-Key Encryption with Keyword Search for Secure Cloud Storage”, IEEE Transactions on Information Forensics and Security , Volume: 11, Issue: 4, Pages: 789 – 798, 2016

[5] Ke Han, Qingbo Li ,Zhongliang Deng, “Security and efficiency data sharing scheme for cloud storage”, Chaos, Solitons and Fractals, Elsevier, Volume 86, Pages 107–116, May 2016

[6] Tao Jiang, Xiaofeng Chena, Jin Li, Duncan S. Wongc, Jianfeng Maa, Joseph K. Liu, “Towards secure and reliable cloud storage against data re-outsourcing”, Future Generation Computer Systems, Elsevier, Volume 52, Pages 86–94, November 2015

[7] Kalpana Batra, Ch. Sunitha, Sushil Kumar, " An Effective Data Storage Security Scheme for Cloud Computing", International Journal of Innovative Research in Computer and Communication Engineering, Volume 1, Issue 4, Pages 808-815, June 2013

[8] Lan Zhou, Vijay Varadharajan, Michael Hitchens, "Cryptographic Role-Based Access Control for Secure Cloud Data Storage Systems", Springer, Security, Privacy and Trust in Cloud Systems, Pages 313-344, 2013

[9] Chuan Yao, Li Xu, Xinyi Huang, Joseph K. Liu, "A secure remote data integrity checking cloud storage system from threshold encryption", Journal of Ambient Intelligence and Humanized Computing, Springer, Volume 5, Issue 6, Pages 857–865, 2014

[10] Sheren A. El-Booz, Gamal Attiya and Nawal El-Fishawy, "A Secure Cloud Storage System Combining Time-Based One-Time Password and Automatic Blocker Protocol", EURASIP Journal on Information Security, Springer, Pages 1-13, 2016

[11] Lan Zhou, Vijay Varadharajan, Michael Hitchens, "Trust Enhanced Cryptographic Role-Based Access Control for Secure Cloud Data Storage", IEEE Transactions on Information Forensics and Security, Volume: 10, Issue: 11, Pages 2381 – 2395, 2015

[12] Wenfeng Wang, Peiwu Li, Longzhe Han, Shuqiang Huang, Kefu Xu, Changgui Yu, and Jin'e Lei1, "An Enhanced Erasure Code-Based Security Mechanism for Cloud Storage", Mathematical Problems in Engineering, Hindawi Publishing Corporation, Volume 2014, Article ID 293214, Pages 1- 8, 2014

[13] Naresh vurukonda, B.Thirumala Rao, "A Study on Data Storage Security Issues in Cloud Computing", Procedia Computer Science, Elsevier, Volume 92, Pages 128 – 135, 2016

[14] Swapnil V.Khedkar , A.D.Gawande, "Data Partitioning Technique to Improve Cloud Data Storage Security", International Journal of Computer Science and Information Technologies, Volume 5, Issue 3 , Pages 3347-3350, 2014

[15] Nirmaljeet Kaur, Harmandeep Singh, "Efficient and Secure Data Storage in Cloud Computing Through Blowfish, RSA and Hash Function", International Journal of Science and Research, Volume 4, Issue 5, Pages 2004-2009, May 2015

[16] Kan Yang, Xiaohua Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing", IEEE Transactions on Parallel and Distributed Systems, Volume 24, Issue 9, Pages 1717 – 1726, 2013

[17] Xinyue Cao, Zhangjie Fu and Xingming Sun, "A Privacy-Preserving Outsourcing Data Storage Scheme with Fragile Digital Watermarking-Based Data Auditing", Journal of Electrical and Computer Engineering, Volume 2016, Article ID 3219042, Pages 1-7, 2016

[18] Yan Zhu, Hongxin Hu, Gail-Joon Ahn and Mengyang Yu, "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, Volume 23, Issue 12, Pages 2231 – 2244, February 2012

[19] Dr. Nedhal A. Al-Saiyd and Nada Sail, "Data Integrity in Cloud Computing Security" Journal of Theoretical and Applied Information Technology, Volume 58, Issue 3, Pages 1-12, 2013.

[20] Heng He, Ruixuan Li, Xinhua Dong, Zhao Zhang, " Secure, Efficient and Fine-grained Data Access Control Mechanism for P2P Storage Cloud", IEEE Transactions on Cloud Computing, Volume: 2, Issue: 4, Pages: 471 – 484, Year: 2014