Privacy and Integrity Verification Model with Decentralized ABE and Double Encryption Storage Scheme

Amrutha Muralidharan Nair[1], Dr. R Santhosh[2]
Research Scholar, Department of Computer Science Engineering, Karpagam Academy of Higher Education, Coimbatore, India[1]
Professor, Department of Computer Science Engineering, Karpagam Academy of Higher Education, Coimbatore, India[2]

***Abstract*—To support a wide range of applications, cloud computing has a variety of services. It has a number of positive acceptance tales as well as a couple of negative ones including security breaches. The versatile usage of cloud services to store Sensitive and personal data in cloud become hesitated by many organizations because of security issues. A new model of relying on a third-party auditor (TPA) has been adopted to improve trust and entice adoption between cloud clients (CC). Hence, we require a dynamic approach to control the privacy and integrity problem that occur across the cloud computing. Decentralized Attribute based encryption techniques and FHE approach is used to overwhelmed the issues. In this proposed scheme, the integrity checking is verified and auditor by the TPA without have any knowledge of the data content and double encryption is performed on the data stored in cloud. the data owner encrypts the data using ABK-XE (Attribute Based Key generation with XOR encryption) technique and send it to tag server whose encrypt the data again using ECEA (Elliptical Curve Elgamal) algorithm and generate the signature and unique ID using SHA-1 algorithm then store the data in Cloud Environment. The proposed algorithm is an integration of auditing scheme with Symmetric key Encryption and Homomorphic Encryption.**

***Keywords*—Cloud computing; data integrity; ABK-XE; ECEA Algorithm; SHA1**

I. INTRODUCTION

Cloud computing [1] is a ground-breaking computing model that has caught the interest of individuals from various fields, including academia and industry. Cloud computing is a large-scale distributed computing paradigm driven by economies of scale in which a pool of abstracted, virtualized, constantly increasing, managed computing power, storage, platforms, and services is supplied on demand to external clients through the Internet [2].

The fundamental component of the cloud is cloud storage, which directs the user's attention to data storage. However, the cloud server is not completely trustworthy, therefore the user confronts several security risks and problems while storing data in the cloud [3]. Data integrity is among the most serious privacy concerns since data hosted on cloud servers is not physically possessed by users and they have no control over it. While some people are prepared to give up their privacy in exchange for the benefits of software services, businesses and governments will never do so [4]. The integrity checking technique provides an efficient approach to examine the data integrity on a cloud server. The client sends an integrity challenge to the cloud server, which creates a proof of the original data. If the evidence passes the verification procedure, the data is shown to be complete. Although numerous systems have been developed to assist data owners in ensuring the integrity of their data. Nonetheless, there are certain difficulties to be addressed. When the integrity verification is taking place, the public verifier is generally interested in the client data and attempts to get it. It is extremely risky for the owner to keep the data secret in light of the aforementioned issues. Encryption of the data is the simple way to make data secure but this method is not at all a good choice for the data user.

To solve the aforementioned issue, this work proposes an effective double encryption technique, and the verification process is done without knowledge of the original data, i.e., it is hidden from the public verifier. The contribution to the paper is listed:

1) A thorough understanding about the elliptical curve encryption.

2) A privacy preservation system with two tier encryption methodology.

3) Detail study about the existing system and usage of bilinear mapping, signing and verifying system.

The remainder of the paper is organized as follows. Section 2 gives a synopsis of the related research. The technique and mathematical formula are described in Section 3. Section 4: Explain the proposed system and its details. Section 5 examines and compares the performance of our system to that of other plans. Section 6 finishes with conclusions.

II. RELATED WORK

Juels et al. [5] suggested the idea of "proof of retrievability (PoRs)". This system not only validates the integrity of data saved in the cloud, but it also assures data retrievability through the use of error-correcting code. Nonetheless, it Private audit is performed.

Yu et al. [6] introduced the ID-based RDIC and its security architecture, which featured protection against a rogue cloud server and privacy from an external auditor with zero knowledge. The RDIC protocol does not leave information about the stored data auditor during the DRIC operation. In the generic group template, the new structure is proven to be

secure against a rogue server and achieves zero knowledge confidentiality against an auditor.

Ateniese et al. [7] developed a "Provable data Possession" (PDP) paradigm to achieve data integrity. The author [8] presented Another scheme scalable PDP scheme, which allows for block adding, updating, and deletion.

Sasikala et al. [9] presented a Remote Data Integrity Checking (RDIC) is critical for developing safe cloud storage.It allows users to validate the integrity of outsourced data without downloading the full file. The author also evaluated and assessed current RDIC techniques based on factors such as integrity testing method, cryptographic model, auditing mode, and data recovery. Finally, it sheds insight on unresolved topics such as research directions in the design of the RDIC procedure.

Yong et al. [10] presented an attribute-based cloud information integrity auditing protocol to reduce key management difficulties. The suggested technique requires significantly less calculation in validating the auditing reaction, resulting in reduced time consumption.

Yannan Li et al. [11] explored the difficult key management problem in cloud data integrity checking by presenting fuzzy identity-based auditing . Author introduced the concept of fuzzy identity-based data auditing, in which a user's identity may be regarded as a set of descriptive properties.

Feng et al. [13] provided a public remote integrity checking technique that protects user identification. This scheme, however, only supports file-level integrity verification.

Yu et al. [14] presented a novel identity-based public RDIC system that protects data privacy. However, the integrity testing of this technique necessitated a significant computational cost.

Tong et al. [15] devised a technique that delivers indistinguishable privacy (IND-privacy) as compared to TPA for both data content and timestamp It build the authenticator with the randomizable structure-preserving signature to connect the content and timestamp in the authenticator and allow efficient timestamp updating (SPS). Furthermore, in the auditing phase, they use the Groth-Sahai proof and range proof to offer IND-privacy and ensure timestamp validity.

Zhang et al. [16] developed an RDIC technique that uses indistinguishability obfuscation to preserve data privacy while improving performance. However, this method is rigid and difficult to implement in practice.

Chen et al. [17] introduced a public verification technique based on algebraic signatures that employed a short bit string compressed by a data block to accomplish efficient integrity verification without comparing to original data. Unfortunately, it is vulnerable to a replay attack. It also does not enable dynamic data update.

III. METHODOLOGY

It is necessary to develop a powerful audit model that solves the issue related to the existing system. The proposed model is made in such a way that the cloud user (CU) can stored the data without heisting, since the data is stored in encrypted form (Fig. 2). The CU can send request to verify the data integrity through TPA. The TPA only send the request and accept the response from the proof-server (PS) without having a computational overhead. Also, it will not provide any information related to use file that is stored in cloud database (CS).

A. Preliminaries

Some of the basic Mathematical approach used in the auditing schemes and in the system model.

1) Bilinear maps: A bilinear mapping $e: G_1 \times G_2 \rightarrow G_t$ where $G_{t\ is}$ the multiplicative cyclic group of prime p with following properties.

a) e is bilinear for all a, b $\in Z_p$

b) e is non-degenerate

c) e is efficiently computable.

2) Elliptic curve Elgamal Algorithm (ECEA): The Elliptical curve [12] with Elgamal algorithm converts the plaintext M to a point P_m on the elliptical curve E. The arithmetic operation on elliptical curve is as follows:

a) Addition of two points: Suppose A = $\{X_a, Y_a\}$and B = $\{X_b, Y_b\}$, where $A \neq B$ that lie on elliptical curve e. the sum of A + B reults a third point C $\{X_c, Y_c\}$ as shown in Fig. 1.

b) Double pointing: Let A = $\{X_a, Y_a\}$ be a point lies on e adding the point A to itself is called Double pointing.

$$P + P = 2P$$

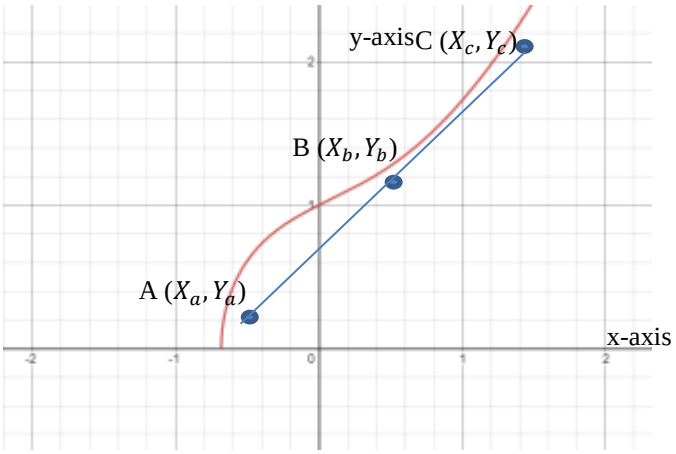

Fig. 1. Elliptical curve of A + B=C.

c) Multiplication: Suppose k is an integer A is a point (X_a, Y_a) then,

$$K_p = \frac{A + \ldots\ldots\ldots + A}{K\ times}$$

So, in this proposed scheme, the computation speed on the ECEA is reduced by using decimal representation of message. So that double pointing and addition operation can be avoided. This scheme involves the following procedures as; Suppose user A and B wishes to communicate with each other. The common parameter known by both users will be p and G, where p is the prime number and G is the base point in the

elliptical curve. Let user A and B choose their private key α_a and α_b over an interval [1, p-1] and generate the public key $\beta_a = \alpha_a G$ and $\beta_b = \alpha_b G$.

The message or file M is divided into n block $m_1, m_2 \dots\dots\dots\dots, m_n$ which is then convert into decimal values $d_1, d_2 \dots\dots\dots\dots, d_n$ respectively such that the multiplication of basepoint and the decimal point transform to a point in E as:

$$p_{d_1} = d_1 G$$

$$p_{d_2} = d_2 G$$

.

.

$$p_{dn} = d_n G$$

Then user A computes the secret key K by multiplying with private key α_a and B's Public key β_b as:

$$K = \alpha_a * \beta_b$$

This k value is added with the Decimal elliptical point to obtain cipher text as:

$$C_1 = p_{d1} + K$$

$$C_2 = p_{d2} + K$$

.

.

.

$$C_n = p_{dn} + K$$

Therefore, the cipher text C= {C_1, C_2……..., Cn}. The plain text is obtaining by subtracting the secret key K` as.

$K` = \alpha_b * \beta_a$ performing the following operation as:

$$p_{d1} = C_1 - K`$$

$$p_{d2} = C_2 - K`$$

.

.

$$p_{dn} = C_n - K`$$

Thus, obtain decimal values is converted to original values $m_1, m_2 \dots\dots\dots\dots m_n$.

Proof,

$$C_1 - K` = C_1 - \alpha_b * \beta_a \qquad (\because K` = \alpha_b * \beta_a)$$

$$\Rightarrow p_{d1} + K - \alpha_b * \beta_a \qquad (\because C_1 = p_{d1} + K)$$

$$\Rightarrow p_{d1} + \alpha_a * \beta_b - \alpha_b * \beta_a \qquad (\because K = \alpha_a * \beta_b)$$

$$\Rightarrow p_{d1} + \alpha_a * \alpha_b G - \alpha_b * \alpha_b G$$

$$(\because \beta_a = \alpha_a G$$

$$\beta_b = \alpha_b G)$$

$$\Rightarrow p_{d1} + \cancel{\alpha_a * \alpha_b G} - \cancel{\alpha_b * \alpha_b G}$$

$$\Rightarrow p_{d1}$$

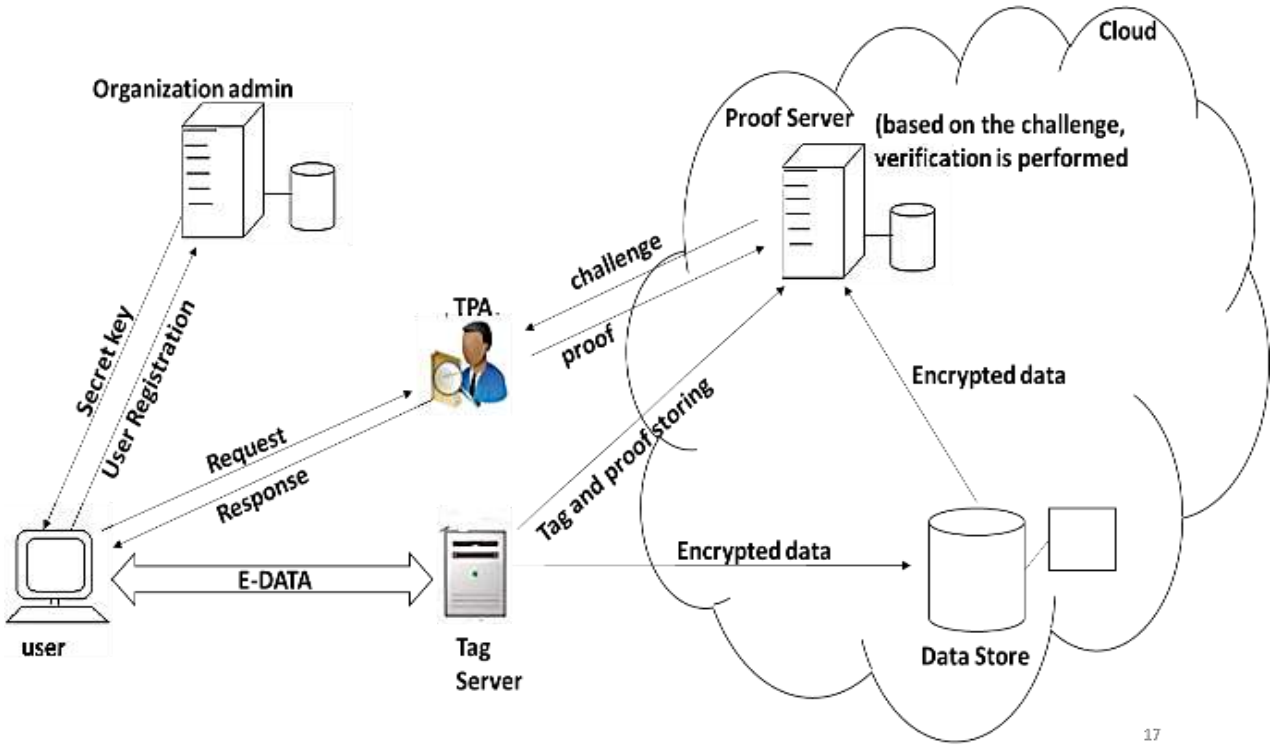

Fig. 2. Proposed system.

IV. PROPOSED SCHEME

The proposed scheme shows a threshold hybrid combination of encryption; this scheme is divided into CU encryption. TagServer processing, PS verification, TPA challenges.

1) CU encryption: In this cloud User (CU) encryption the file or data that he/she wishes to store in cloud by using it attribute UID and PW. This encryption/decryption is divided into two phases:

a) Key generation process: This generates a symmetric key based on user attributes:

i. User will provide its password as the seed value, that will be convert into byte form as:

$$\text{byte b[]} = \text{byte(PW)}$$

ii. Calculate the sum as:

$$B = \sum_{i=0}^{b.length} b(i)$$

iii. Applying log function and convert the value to a factor of 64

$$k_b = \log B$$

iv. Using the bilinear paring the b(i) will be multiplied by K so that we obtain the symmetric key array$A[i]$,

$$for\ each\ i = 0\ to\ b.length$$
$$A[i] = k_b * b(i) mod\ 64$$
$$end$$

b) Encryption process: The generated key $A[i]$ will be XoR with the file $F = \{f_1, f_2, \dots\dots\dots f_n\}$ as follows.

i. The file F is divided into n blocks of size 64 bits as $= \{f_1, f_2, \dots\dots\dots f_n\}$, the last is padded with zeros if required.

ii. Each block is XoR with the symmetric key as:

$$C[i] = \sum_{i=1}^{n} f[i] * A[i]$$

iii. This obtain C(i) will be combined together to form the encrypted file $E_A(F)$

The above algorithm is shown in Fig. 3.

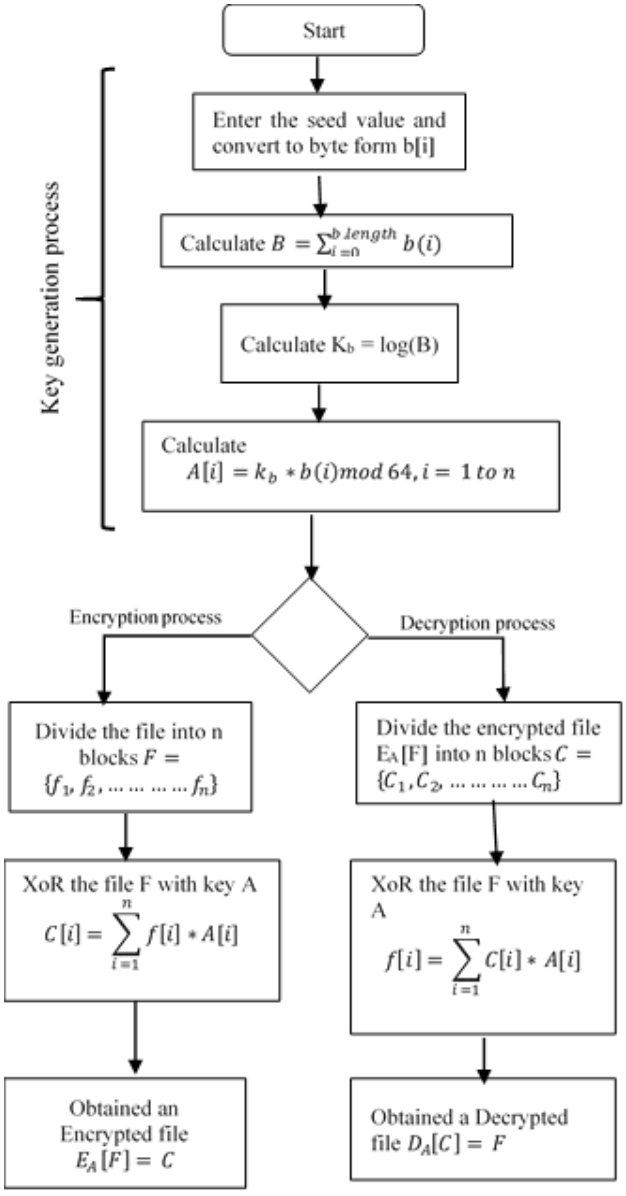

Fig. 3. User side encryption and decryption process.

2) TagServer Processing (TSP): TSP plays a curial role in the scheme. Here the encrypted file $E_A[F]$ received from the user undergoes ECEA encryption process and obtain the encrypted form as $C^* = Encrpyt_{ECEA}(E_A[F]$. Then TSP calculates the Signature of the encrypted file C^* by using SHA-1 algorithm. Thus, signature digest is of the form 160-bit represented as $Sig_{C^*} = MD(C^*)$. TSP transfers the encrypted file C^* to cloud data store; also store the signature along with the unique file Id ($Sig_{C^*}| |F_{id}$). TSP also send its public key and file Id to cloud user so that the CU can decrypt the file directly from the data store ($F_{id}| |\beta_b| |T_1$) as shown in Fig. 4.

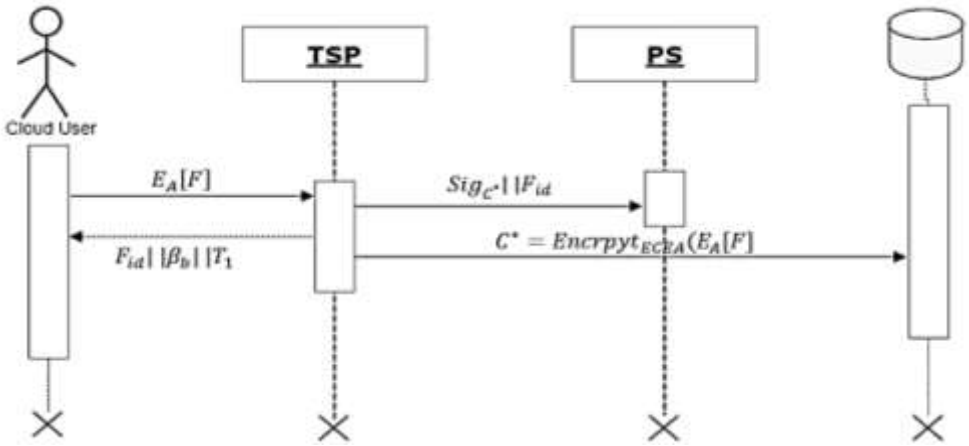

Fig. 4. Working of TSP.

3) TPA challenges and Proof Server verification: When the cloud user CU require to verify the file, it sends a challenge request to TPA as $ChalR_Q = F_{id} \,||\, ID_{cu}|| \, |T_2$. The auditing of the file is performed by sending a request to the TPA by the cloud user CU. The receive request $ChalR_Q$will be transmitted to proof Server PS.

The PS Retrieve the file from the data store using the F_{id}, compute the signature of the file $Sig^{|}_{C^*}$ using SHA-1 Algorithm and compare the signature store by TSP (Sig_{C^*}) and new compute signature $Sig^{|}_{C^*}$. This comparison declared that the file stored in the data store is altered or not. This response Res_Qwill be intimated to TPA which then forward to CU with the assurance of it file integrity is maintained.

Here the TPA and PS doesn't have direct access to the original file F so the semi trust issues is solved by using this scheme also it provides high assurance of data integrity in the cloud environment as shown in Fig. 5.

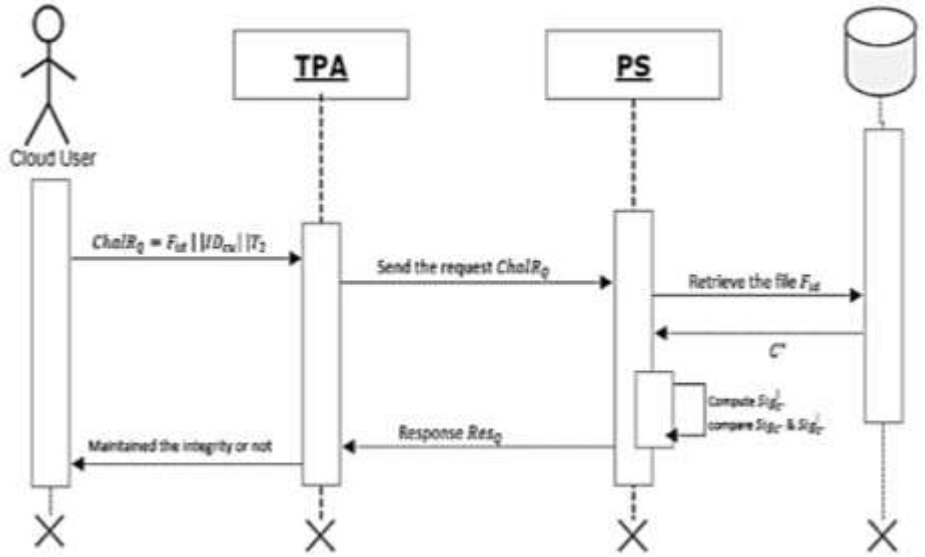

Fig. 5. Working process of TPA and PS.

V. Performance Evaluation

The proposed scheme performance is summarized as follows.

A. Computation Cost

The above specified algorithm contains operations such as multiplication, paring operation, addition and subtraction operation with the time Tm, Tp, Ta, Ts on the curve G respectively. The Time taken for other operation such as hash function and logarithm is were less so the computation cost for such operation can be negligible cost. Suppose the cloud user contain m blocks in total, cl challenge is passed to TPA for checking the integrity of data block in cloud. The computation cost in TagServer algorithm for encrypting to run m times to encrypted the m blocks and generated the signature is $(T_m(m+1)) + mT_a$. The computation cost at TPA to transfer the challenge is negligible cost. The computation cost at the proof Server is $(cl+1)T_m + clT_a$. The verification cost of the proposed scheme is $T_P + (cl+1)T_m + clT_a$. The decryption algorithm computation cost is $(T_m(m+1)) + mT_s$. The below Table I shows the computation cost of our proposed model with Scheme [13] and [15].

TABLE I COMPUTATION COST COMPARSION

COST	SCHEMES		
	[13]	[15]	*Proposed model*
Encryption cost	$2T_{ex} + T_m$	$2T_{exp} + T_m$	$(T_m(m+1)) + mT_a$.
Proofgenerartion	$2T_p + (cl+1)T_{ex} + clT_m$	$(cl+1)T_{ex} + (cll-1)T_m$	$(cl+1)T_m + clT_a$
Verification process	$clT_p + (cl+1)T_{ex}$	$2T_P + (cl+2)T_{ex} + (cl+1)T_m$	$T_P + (cl+1)T_m + clT_a$.
Decryption process	$2T_{ex} - 1 + T_m$	$2T_{exp} - 1 + T_m$	$(T_m(m+1)) + mT_s$.

B. Experimental Results

To evaluate the efficiency of the proposed model, the model and compared model is implemented in Intel i5 core with windows 10 operating system, 8GB RAM and 1TBB hard disk. All the experiment was carried out by using different User of cloud environment for which the minimum configuration of 1 CPU is to make the research cost effective. It is simulated using CloudSim with NetBeans framework (Java language).

The main section of the proposed system is the encryption of encrypted file and generation of signature done by the tag server; verification of data integrity performs by the proof server. This scheme is evaluated by considering variable size block from 1KB to 100KB, as a result its encryption time, siggeneration time, sigverification time is obtained in millisecond(ms) as shown in the Table II. This result show that as the block size increases the encryption, signature generation or verifying time is less than the time taken for the 1 KB block. The result of the proposed scheme is shown in Fig. 6 and 7.

The efficiency of algorithm used for proof generation and verification is evaluated. Furthermore, we implemented the scheme [13] and [15] under the same experiment setting and make an efficiency comparison with each other. In the experiment the number of challenges in the system is increased. Thus, the result which is illustrated in Fig. 8 shows the efficiency of the algorithm when the challenges has increased from 100- 1000. Fig. 8 shows our scheme is little more efficient than the scheme [13] and [15] in proof generation. From Fig. 9, the verification purpose our scheme requires 5 second for 1000 challenge block and scheme [15] and [13] require almost 6.1 second for the same block.

TABLE II PROPOSED SYSTEM TIME COMPLEXITIES

Block size	*TIME*		
	Encrypting time	*Proof generation*	*Proof verification*
1 KB	185417	1138	1690
10 KB	183250	938	1234
20 KB	182189	930	1162
30 KB	180123	696	1088
40 KB	174960	464	1486
50 KB	152192	264	1424
60 KB	150215	440	1386
70 KB	147356	252	1028
80 KB	142523	232	1296
90 KB	135661	212	388
100 KB	128114	212	258

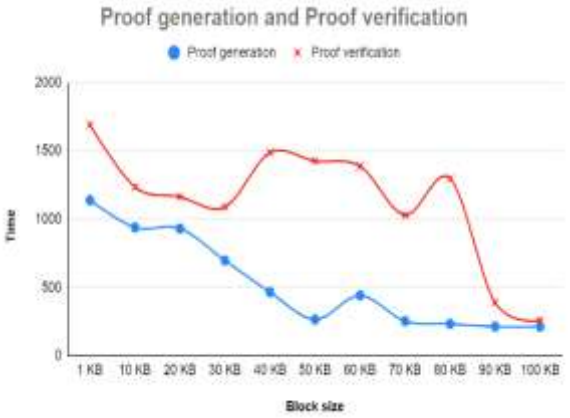

Fig. 6. Proof generation and verification time consumption.

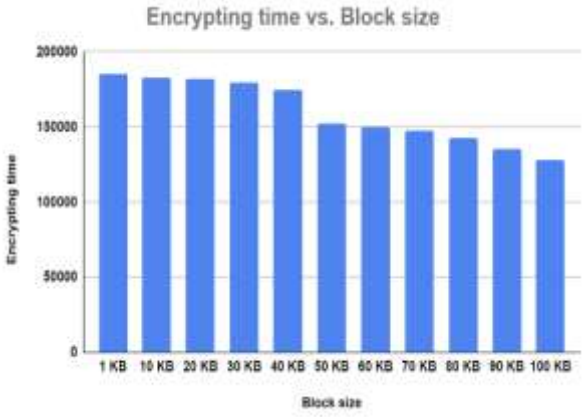

Fig. 7. Time consumption for encryption.

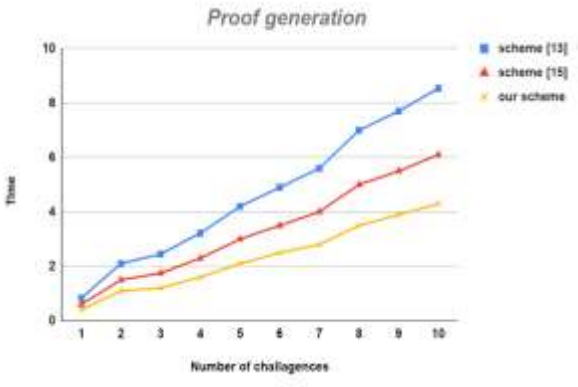

Fig. 8. Time consumption in proof generation process.

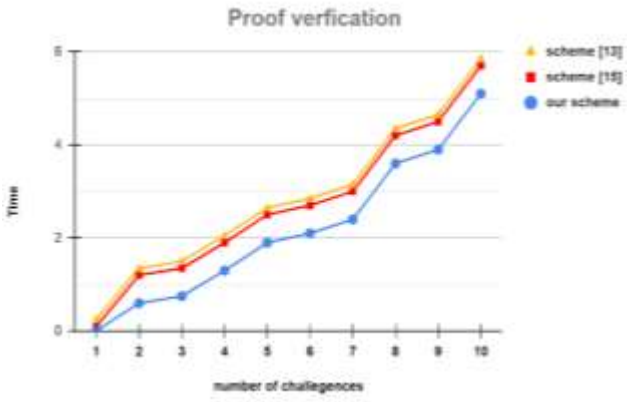

Fig. 9. Time consumption in verification process.

VI. CONCULSION

In cloud environment, data privacy is very much essential one. In this paper, we have proposed a technique which is a combination of symmetric and asymmetric encryption. The proposed scheme is a combination of XoR encryption and ECEA encryption with SHA-1 algorithm, which is suitable for integrity auditing of data stored in cloud computing. The existing schemes are meant to provide integrity maintenance for numerous data stored in cloud, but it doesn't provide dynamic data operation, data are visible to TPA totally.

The proposed scheme preserved the confidentially to the file attributes and simplified the key management system in the traditional cloud facts. This scheme efficiently and effectively supports auditing tasks such as guaranteeing the TPA integrity, secure storage, overall control in the system. The result of the implementation which is shown above define that the proposed scheme provides 0.1% more security than the existing protocols. This paper explains the complete details about the construction, implementation and experimental results of the proposed model.

REFERENCES

[1] R.Buyya, C.S. Yeo, S. Venugopal, J.Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platform: Vision, Hype, and Reality for Delivering Computing as the %th utility," Future Gener. Comp. Sy., vol. 25, no. 6, pp. 599-616, 2009.

[2] Jiguo Li, Hao Yan, Yichen Zhang ,"Certificate less public integrity checking of group shared data on cloud storage," IEEE Transactions on Services Computing,, 2018.

[3] M. Ali, S.U. Khan, A.V. Vasilakos, "Security in Cloud Computing: Opportunities and Challenges," Inf. Sci, vol. 305, no. 1, pp. 357-383, 2015.

[4] S. Suganya, "Improving Cloud Security by Enhancing Remote Data Integrity Checking Algorithm," Innovations in Power and Advanced Computing Technologies (i-PACT) IEEE, 2017.

[5] A. Juels and B. J. Kaliski, "PORs: Proofs of Retrievability for Large Files," in *In Proceedings of the 14th ACM Conference on Computer and Communications Security*, Alexandria, 2007.

[6] Yong Yu, Man Ho Au, Giuseppe Ateniese, Xinyi Huang, Willy Susilo, "Identity-Based Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage," IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 767-778, 2017.

[7] Ateniese, Giuseppe & Burns, Randal & Curtmola, Reza & Herring, Joseph & Kissner, Lea & Peterson, Zachary & Song, Dawn, "Provable Data Possession at Untrusted Stores," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2007.

[8] G.Ateniese,R.D.Pietro,L.V.mancini ,and G.Tsudik., "Scalable and Efficient Provable Data Possession," in *Proc. Fourth Int'1 Conf. Security and Privacy in COmmun. Netw.(SecureComm'08)*, 2008.

[9] Bindu, C. Sasikala and C. S., "A study on remote data integrity checking techniques in cloud," in *International Conference on Public Key Infrastructure and its Applications (PKIA)*, 2017.

[10] Yong Yu, Yannan Li, Bo Yang, Willy Susilo, Guoming Yang and Jian Bai, "Attribute-Based Cloud Data Integrity Auditing for Outsourced Storage." *IEEE Transaction on Emerging Topics in Computing* vol. 8, no. 2, pp. 377-390, 2017.

[11] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni and K. R. Choo, "Fuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 72-83, 2019.

[12] A. Odlyzko, "Discrete Logarithms in Finite Fields and Their Cryptographic Significance," In Workshop on the Theory and Application of of Cryptographic Techniques; Springer: Berlin/Heidelberg, Germany, 1987.

[13] Y. Feng, G. Yang and J. K. Liu, "A new public remote integrity checking scheme with user and data privacy," *International Journal of Applied Cryptography (IJACT)*, vol. 3, no. 3, pp. 196 - 209, 2017.

[14] Yong Yu, Man Ho Au, Giuseppe Ateniese, Xinyi Huang, Willy Susilo. "Identity-Based Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage." IEEE Transactions on Information Forensics and Security, vol.12, no. 4,pp: 767-778,2017.

[15] Tong Wu, Guomin Yang, Yi Mu, Rongmao Chen, Shengmin Xu,"Privacy-enhanced remote data integrity checking with updatable timestamp",Information Sciences,vol. 527, pp:210-226,2020.

[16] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu and X. Zhang, "Efficient Public Verification of Data Integrity for Cloud Storage Systems from Indistinguishability Obfuscation," in IEEE Transactions on Information Forensics and Security, vol. 12, no. 3, pp. 676-688, March 2017.

[17] Chen, L. Using Algebraic Signatures to Check Data Possession in Cloud Storage. Future Gener. Comput. Syst., vol. 29, pp:1709–1715, 2013.