Volume 9, Issue 5
October 2020

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X



# Attribute based Integrity Auditing using Elliptical Curve Modified RSA Algorithm for Cloud Data

**[1]Yogita Sharma and [2] Neetesh Kumar Gupta**

[1]Department of Computer Science and Engineering, TIT, Bhopal (M.P), India

[2]Department of Computer Science and Engineering, TIT, Bhopal (M.P), India

E-mail: [1]yogitasharmatit@gmail.com

## ABSTRACT

Cloud computing is a vast area within which large amounts of data are exchanged through cloud services and has fully grown with its on-demand technology. as a result of these versatile cloud services store sensitive data on cloud storage servers, you need to dynamically control a number of problems: security, privacy, data privacy, data sharing, and integrity across cloud servers. moreover, the legitimacy and control of data access should be maintained in this extended environment. So, one of the most important concepts of cryptographic techniques in cloud computing environment is Attribute Based Encryption (ABE). In this research work data auditing or integrity checking is an area of concern for secure cloud storage. In data auditing approach, an auditor inspects and verifies the data file integrity without having any knowledge about the content of file and sends the verification report to the data owner. In this research, Elliptical Curve Modified RSA (ECMRSA) is proposed along with Modified MD5 algorithm which is used for attribute based cloud data integrity verification, in which data user or owner uploads their encrypted data files at cloud data server and send the auditing request to the Third Party Auditor (TPA) for verification of their data files. The Third Party Auditor (TPA) challenges the data server for integrity of data files in behalf of data owners. After verification of integrity of data file auditor sends the audit report to the owner. The proposed algorithm integrates the auditing scheme with public key encryption with homomorphic algorithm which generates digital signature or hash values of data files on encrypted files.

**Keywords:** *Cloud Computing, Data Integrity, Data Auditing, Data sharing, Security, ABE*

## 1. INTRODUCTION

With the development of information technique, more and more data are generated and need to be dealt with. Due to the finiteness of computing capabilities and storage space of personal computers, it is urgent for people to find a new way to solve the problems arising from a large number of data. In recent years, the emergence of cloud computing satisfies people's needs in this area. Cloud computing is a large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet [1].

Cloud storage which is an important component of cloud computing brings great convenience to users. It is an emergent scheme that puts the storage resources into the cloud and allows users to store or obtain data. Users will obviate the costs of building and maintaining a private storage infrastructure if they outsource their data in the cloud [2]. They don't have to worry about the lack of computing capabilities and storage space. Besides, they can conveniently access their data anytime and anywhere. Although it is obvious that cloud storage has a lot of benefits, security issues still exist [3].

Data integrity is one of the most significant security concerns since the data outsourced in cloud servers are not physically possessed by the users and the control of data is not in the hands of them. While some users may be willing to sacrifice their privacy for benefits brought from software services, enterprises and government organizations will never do that [3]. As a primitive, data integrity checking guarantee that users' data keep intact on the cloud servers. On other kind of distributed computing systems, signature, digest, replica or other methods are used to guarantee data integrity [4-5].

This research work is focused on authorized third party auditing scheme in order to audit data integrity checking at cloud servers. Now-a-days data integrity checking at data server is an effective area of research work. This research work focuses on study of different protocols that are intended towards data integrity checking at cloud servers [6-8].

The main contributions of this research work are as follows:

- the various system models and threat models for outsourcing information in cloud are mentioned.
- A comparative analysis of these protocols on the idea of security strategies, storage overheads, computation prices, communication cost, etc.
- The challenges for actualizing an robust data integrity auditing protocol are highlighted.

The objective of the research work are as follows:

- To propose an attribute-based cloud data integrity auditing, where users can choose some set of attributes to generate secret keys and upload files to cloud server.
- Moreover, the data owners can specify a Third Party Auditor (TPA) who are able to check the integrity of the outsourced data [9].

## 2. RELATED WORK

Yong Yu et al. [2] proposed the ID-based RDIC and its security model, including security against a malicious cloud server and zero knowledge privacy against an external auditor.

Volume 9, Issue 5
October 2020

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X



The proposed ID the RDIC protocol does not leave information about the stored data auditor during the DRIC process. The new construction is demonstrated protected against the malicious server in the generic group template and obtains the confidentiality of zero knowledge against an auditor. Extensive the safety analysis and implementation results prove it the proposed protocol is proven safe and practical in the applications of the real world.

Sasikala [5] discussed that Remote Data Integrity Checking (RDIC) is crucial for implementing secure cloud storage. It enables the users to check the integrity of the outsourced data without downloading the entire data. This research presents a detailed study on RDIC protocols in a cloud environment and discusses the taxonomy of RDIC protocols such as Provable Data Possession (PDP), Proof of Retrievability (POR), Proof of Ownership (POW) and ID-based RDIC protocols. Author also analyzed and compared the existing RDIC approaches based on the parameters such as integrity checking method, cryptographic model, auditing mode and data recovery. Finally, it throws light on the open issues such as research directions in designing RDIC protocol.

R.Swathi and T.Subha [7] proposed an approach named enhancing data storage security in cloud using Certificate less public auditing scheme which is used to generate key value. Key Generation Center (KGC) will generate only the partial key so that at any case it will not compromise user's private key. Private & public key is generated based on the partially generated private key by the KGC and to check the cloud data reliability of the user's uploads the data in server and then during the auditing of the reliability of data is checked. Once after checking it then sends the report to the users'. To confirm the data reliability during the auditing process & the server generates the proof and randomly selects the blocks. The TPA then authenticates the proof against cloud server & the auditing result to the user.

Yong Yu et al. [8] proposed an attribute based cloud information integrity auditing protocol to simplify the key management problems. the proposed method have much less calculation in verifying the reaction of auditing and as a result reason less time consumption.

Yannan Li [10] discussed the complex key management challenge in cloud data integrity checking by introducing fuzzy identity-based auditing, the first in such an approach, to the best of our knowledge. More specifically, author presented the primitive of fuzzy identity-based data auditing, where a user's identity can be viewed as a set of descriptive attributes. They formalize the system model and the security model for this new primitive. We then present a concrete construction of fuzzy identity-based auditing protocol by utilizing biometrics as the fuzzy identity. The new protocol offers the property of error-tolerance, namely, it binds with private key to one identity which can be used to verify the correctness of a response generated with another identity, if and only if both identities are sufficiently close. Author proved the security of designed protocol based on the computational Diffie-Hellman assumption and the discrete logarithm assumption in the selective-ID security model.

It is noted that cloud servers are not completely trusted servers or platform. At cloud data server if there is any case of accidental damage of the data file of owners which may leads to permanent damage of data files of cloud users. In such case data files are recovered from backups at data server by using rollback techniques. After applying backup, data files are found in previous state of data files. So, in such condition data integrity checking is required in order to update the data owner about their outsourced data in order to make sure the cloud servers store their data correctly. So, it is necessary to design less complex data integrity mechanism with high efficiency.

## 3. METHODOLOGY

It is necessary to develop a powerful public audit protocol that solves the problem of the existing audit system. The proposed gadget is designed to periodically or on demand monitor the accuracy of cloud data through TPA, without finding all the facts or imposing an extra burden online to cloud users and cloud servers. Ensures that no problems are transmitted to TPA during the control method. The integrity of statistics, integrity and confidentiality of archived records continues.

The proposed system consists of three basic units; data owner, cloud server storage and TPA. The owner or user of the data is responsible for splitting the file into blocks, encrypting an encryption algorithm, for each hashing and for hash concatenation on them. The cloud server is used to store blocks of encrypted documents. When the client or data owner requires data validation at the TPA, it immediately requests the encrypted data from the cloud server.

After receiving the data, it generated the hash value for each block of encrypted files. It uses the same hashing algorithm which was used by client. If they match with each other it means that the data is intact and data is not being tampered by any outsider or attacker. If it does not match then it indicates that the data integrity has been affected or tampered. The result for the data integrity check is provided to the data owner. Figure 1 shows the architecture for proposed parametric approach.

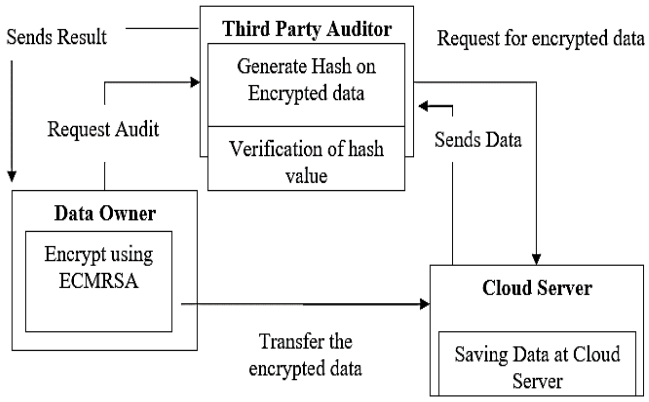


Fig. 1. Proposed Architecture

In order to address the issue mentioned above, the concept of cloud data integrity auditing was presented, namely Provable Data Possession (PDP). PDP is a probabilistic detection protocol which employs randomly sampled data blocks rather than the entire file to perform cloud data integrity checking, which is more efficient than the deterministic auditing protocols, especially for large files.

As part of this research, cloud data integrity assessment is performed through the introduction of attribute-based cloud

Volume 9, Issue 5
October 2020

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X



data validation, which allows users to upload files to the cloud using certain attributes and designate attributes. listeners to verify the integrity of the files. Variable attributes are used to generate the private key and their performance is evaluated in a list of variable attributes.

Step 1: Input data

Step 2: Encrypt data using Elliptical Curve Modified RSA (ECMRSA).

Step 3: Divide encrypted data file into blocks of different sizes i.e. 10KB upto 100KB block size

Step 4: Generate Digital Signature of each block using M-MD5 algorithm

Step 5: Send the encrypted data blocks to the cloud data server to store these blocks.

Step 6: If user wants to check integrity of stored data or wants to audit data

{

Data owner send request to TPA

Cloud Server sends data blocks to TPA

TPA generates hash code on these blocks and send the audit report to data owner.

}

Step 7: Exit

### *A. Elliptical Curve Modified RSA (ECMRSA) Encryption and Decryption Algorithm*

Find point on a curve, $P_b$ such that $P_b > P_T$ and $P_b \in Ep$

Let $E_p$ be an elliptic curve equation over a finite field

$Ep : y^2 = x^3 + ax + b \bmod p$

Where, a and b are constant

P= prime number

Coordinates $(x, y) \in E_p$ follows certain additive abelian properties.

$e = d * P_b$

Where, d = The random number that we have selected within the range of (1 to n-1).

e' is the public key and 'd' is the private key.

For encryption, $C_T = P_T^e \bmod n_1$

$P_1$ = prime number less than $n_1/2$;

$Q_1$= prime number less than $n_1/2$;

$n = P_1 * Q_1$;

Where, $n_1 = (x_c, y_c) - n_b P_b$

$(x_c, y_c)$= One of the point in elliptic curve after point addition of $P_T$ and $kP_b$.

k= random integer between 1 and n − 1 where n is the cyclic order of an elliptic curve over finite field.

nb= Receiver's attributes (taken from user)

For decryption, $PT = CT^d \bmod n$

An attribute-based cloud data integrity audit log includes four entities:

Key Generation Center (KGC): KGC manages the user's private key generation based on its group of attributes.

Cloud users

Cloud server

TPA: TPA is a third party designated to verify the integrity of cloud data on behalf of cloud users after a verification request.

The details of an attribute based cloud data integrity auditing protocol are described below:

a. A cloud user passes an attribute set to KGC to request a secret key.
b. KGC generates a secret key for the user with the password and user attributes.
c. The cloud user generates the metadata of the encrypted file, i.e. signature. The user then downloads the encrypted file with the corresponding signature in the cloud.
d. Upon receipt of the verification request, TPA and the cloud server execute a query response protocol based on a homomorphic algorithm to verify the stored file.

This primitive includes the following four algorithms:

e. Setup (k): This algorithm generates the main MK key and the public PK parameters.
f. Extract (MK, A): This is an algorithm that uses a master MK key and an input A attribute set to create the secret SKA key for the user.
g. Character (PK, SKA, M): This is an algorithm that uses the public PK parameter, a secret SKA key and an input M message. There is a data file signature.
h. Verification (PK, B, M): This is a deterministic algorithm that uses the public parameter PK, a set of attributes B, the message M and its presumed signature as input. Returns 1 or 0 to indicate that the signature is valid or not.

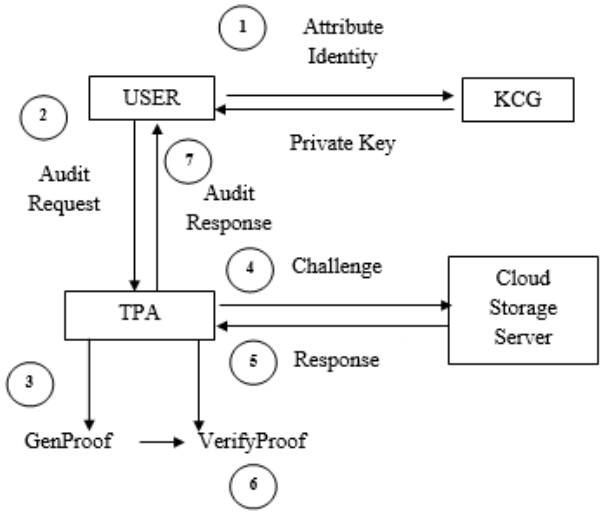


Fig. 2. Proposed Architecture

The protocol architecture is based in three components:

(i) clients

(ii) storage services in the cloud

(iii) an integrity check service

The protocol includes two different execution processes. The former is called a "file storage process" and is executed on demand with the client as a boot device. The second is the "verification process", which is instantiated by an integrity check service and runs continuously to verify a cloud storage service.

Our public auditing system can be constructed from the above auditing scheme in two phases, Setup and Audit.

**Setup:** The user initializes the public and secret parameters of the system by executing KeyGen, and preprocesses the data file F by using SigGen to generate the verification metadata. The user then stores the data file F at the cloud server, deletes its local copy, and publishes the verification metadata to TPA for later audit. As part of pre-processing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.

*A.Algorithm is used to KeyGen Process*

Volume 9, Issue 5
October 2020

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X



i. Find point on a curve, $P_b$ such that $P_b > P_T$ and $P_b \in Ep$
Let $E_p$ be an elliptic curve equation over a finite field
Ep : $y^2 = x^3 + ax + b \bmod p$
Where, a and b are constant
P= prime number
Coordinates (x, y) $\in E_p$ follows certain additive abelian properties.

ii. $e = d * P_b$
Where, d = The random number that we have selected within the range of (1 to n-1).
e' is the public key and 'd' is the private key.

### *B. Algorithm used to SigGen Process*

The M-MD5 algorithm works in the following steps:
The MD5 message digest hashing algorithm processes data in 512-bit blocks, broken down into 16 words composed of 32 bits each. The output from MD5 is a 128-bit message digest value.

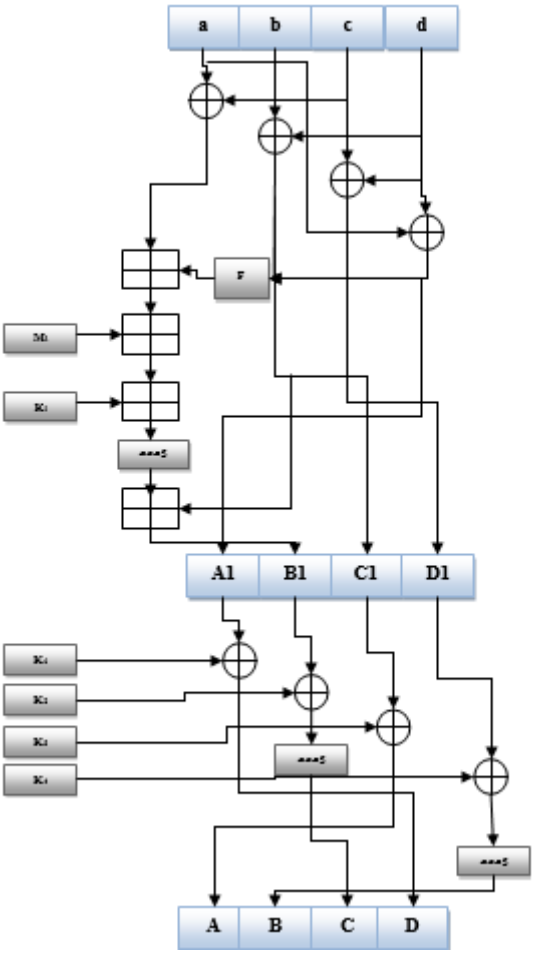


Fig. 3. Process block of M-MD5

The steps are as follows as :
Computation of the MD5 digest value is performed in separate stages that process each 512-bit block of data along with the value computed in the preceding stage. The first stage begins with the message digest values initialized using consecutive hexadecimal numerical values. Each stage includes four message digest passes which manipulate values in the current data block and values processed from the previous block. The final value computed from the last block becomes the MD5 digest for that block. Figure 3 shows the proposed MD5 algorithm flowchart in which $M_i$ is the expanded message word of round i. Ki is the round constant of the round i. ⊞ denotes addition modulo. F is a non-linear function that varies. The original key is divided into 4 parts as $K_1$, $K_2$, $K_3$ and $K_4$.

Audit: The TPA sends a monitoring message or a request to the cloud server to ensure that the cloud server has correctly stored the F data file at the time of the audit. The cloud server will get a response message from a function of the F stored data file by running GenProof. Using verification metadata, the TPA verifies the response through VerifyProof.

## 4. RESULT ANALYSIS

To evaluate the proposed algorithm, it is concentrated on four indications of performance i.e. Block Generation Time, Encryption Time, GenProof Time and VerifyProof Time.

Block Generation Time: It is the time taken to generate data blocks.

Encryption Time: It is the time taken to encrypt data blocks.

GenProof Time: It is the time taken to generate digital signature or hash value of each encrypted data blocks.

VerifyProof Time: It is the time taken to verify the challenge generated for integrity check.

### *A. Performance Evaluation*

In this section, we report the performance of the proposed protocol. In our implementation, all the algorithms are conducted on a Win 10 64-bit laptop with Intel Core i5 processor and an 8 GB Hard-disk. The research work is simulated by using cloudsim using netbeans platform.

In the first part, we present the time consumption evaluation of Extract, Genproof as well as verifyproof algorithms. The result analysis is shown in Table I which is evaluated by using variable size block of 1MB file. The block size varies from 1KB to 100KB with the increment of 10KB The simulation is performed on encrypted data file. So, the table illustrates four different time complexities i.e. block generation time, Encryption time, genproof time as well as verifyproof time.

Table I: Performance Evaluation of Proposed Algorithm

| Block Size | Block Generation Time (in ms) | Encryption Time (in ms) | GenProof Time (in ms) | Verify Proof Time (in ms) |
|---|---|---|---|---|
| 1 KB | 690 | 183250 | 569 | 845 |
| 10 KB | 159 | 152192 | 469 | 617 |
| 20 KB | 141 | 185417 | 465 | 581 |
| 30 KB | 134 | 150215 | 348 | 544 |
| 40 KB | 133 | 182189 | 232 | 743 |
| 50 KB | 127 | 147356 | 132 | 712 |
| 60 KB | 126 | 180123 | 220 | 693 |
| 70 KB | 127 | 174960 | 126 | 514 |
| 80 KB | 123 | 128114 | 116 | 648 |
| 90 KB | 122 | 142523 | 106 | 194 |
| 100 KB | 109 | 135661 | 106 | 129 |

Volume 9, Issue 5
October 2020

ITEE Journal
Information Technology & Electrical Engineering

ISSN: - 2306-708X



As can be seen from Fig. 4, the time cost of the extract algorithm exhibits a linear growth with the maximum number of attributes m in the system.

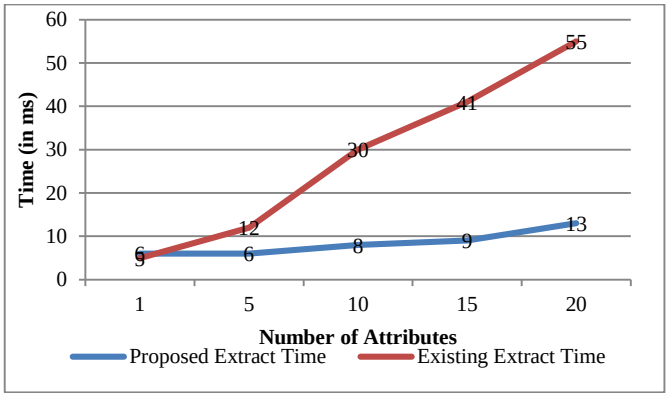


Fig.4. Time Consumption for Extract Algorithm

As can be seen from table II and Fig.5, the time cost of the Genproof algorithm with variable block size. 1MB data file is divided into 10KB file blocks upto 100KB file blocks. From the graph it has been noticed that as the number of block increases the genproof time complexity reduces.

Table II: Time consumption for GenProof algorithm of 1MB file

| Block Size | Proposed GenProof Time (in ms) | Existing GenProof Time (in ms) |
|---|---|---|
| 10 KB | 569 | 4000 |
| 20 KB | 469 | 3000 |
| 30 KB | 465 | 2000 |
| 40 KB | 348 | 1600 |
| 50 KB | 232 | 1500 |
| 60 KB | 132 | 1400 |
| 70 KB | 220 | 1300 |
| 80 KB | 126 | 1000 |
| 90 KB | 116 | 1000 |
| 100 KB | 106 | 1000 |

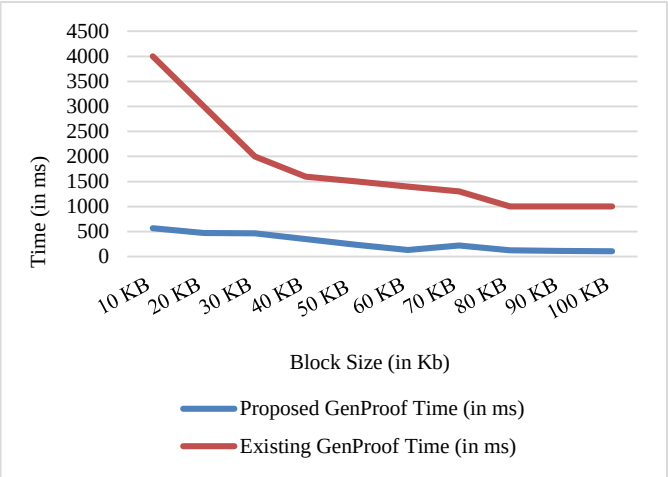


Fig.5. Time consumption for GenProof algorithm of 1MB file

Table III: Time consumption for VerifyProof algorithm of 1MB file

| Block Size | Proposed VerifyProof Time (in ms) | Existing VerifyProof Time (in ms) |
|---|---|---|
| 10 KB | 1845 | 20000 |
| 20 KB | 1617 | 15000 |
| 30 KB | 1581 | 13000 |
| 40 KB | 1544 | 12000 |
| 50 KB | 1743 | 11000 |
| 60 KB | 1712 | 10000 |
| 70 KB | 1693 | 10000 |
| 80 KB | 1514 | 8000 |
| 90 KB | 1648 | 8000 |
| 100 KB | 1194 | 8000 |

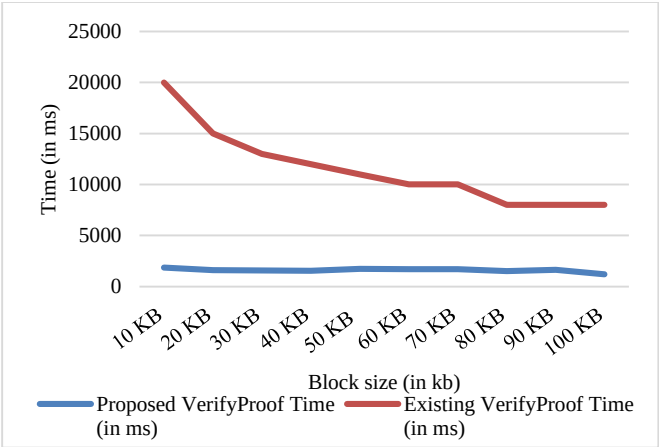


Fig.6. Time consumption for VerifyProof algorithm of 1MB file

As can be seen from table III and Fig. 6, the time cost of the Verifyproof algorithm with variable block size. 1MB data file is divided into 10KB file blocks up to 100KB file blocks. From the graph it has been noticed that as the number of block increases the verifyproof time complexity reduces.

## 5. CONCLUSION

As mentioned in this research, most existing protocols are meant to provide integrity management for numerous data storage systems, however don't totally support dynamic data operations, public responsibility and confidentiality. the basic needs that an integrity protocol should meet are mentioned. once designing a data integrity validation protocol, it's necessary to concentrate to the fact that it's efficient, safe and complies with the essential needs.

The proposed protocol has preserved the confidentiality of statistics file attributes, simplifying the key management problem in traditional cloud facts control structures. This implementation of the proposed system illustrates the practicality and effectiveness of the system.

In this research, Elliptical Curve Modified RSA (ECMRSA) is proposed along with Modified MD5 (MMD5) algorithm which is used for attribute-based cloud data integrity verification. The Third Party Auditor (TPA) challenges the data

server for integrity of data files in behalf of data owners. After verification of integrity of data file auditor sends the audit report to the owner.

The proposed algorithm integrates the auditing scheme with public key encryption with homomorphic algorithm which generates digital signature or hash values of data files on encrypted files. As block sizes increase, processing costs increase both in GenProof and in VerifyProof, but provide a more efficient result in terms of existing work.

## REFERENCES

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

[2] 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, Volume: 12, Issue 4, 2017, pp. 767 - 778.

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

[4] T.Subha "Efficient Privacy Preserving Integrity Checking Model for Cloud Data Storage Security", International Conference on Advanced Computing (ICoAC), IEEE, January 2017.

[5] C. Sasikala "A Study on Remote Data Integrity Checking Techniques in Cloud", International Conference on Public Key Infrastructure and its Applications (PKIA), IEEE, Nov. 2017.

[6] Samundiswary. S "Public Auditing for shared data in cloud with safe user revocation", International conference of Electronics, Communication and Aerospace Technology (ICECA), IEEE, 2017.

[7] R.Swathi and T.Subha, "Enhancing Data Storage Security in Cloud using Certificateless Public Auditing", International Conference on Computing and Communications Technologies (ICCCT), IEEE, February, 2017, pp. 348-352.

[8] Yong Yu, Yannan Li, Bo Yang, Willy Susilo, Guoming Yang and Jian Bai, "Attribute-Based Cloud Data Integrity Auditing for Secure Outsourced Storage". IEEE Transactions on Dependable and", IEEE Transaction on Emerging Topics in Computing, Vol. 14, No. 8, 2017.

[9] Khalid El Makkaouia, Abderrahim Beni-Hssaneb, Abdellah Ezzatia, Anas El-Ansari, "Fast Cloud RSA Scheme for Promoting Data Confidentiality in the Cloud Computing", Procedia Computer Science, Volume 113, 2017, pp. 33–40.

[10] Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, K-K. R. Choo. "Fuzzy Identity-Based Data Integrity Auditing for Reliable Cloud Storage Systems", IEEE Transactions on Dependable and Secure Computing, Volume 16, issue 1, 2017, pp. 72-83.