[1] Srividya B V *
[2] Smitha Sasi
[3] Soumya S
[4] Harikeerthan M K

Securing the Cloud: A New Horizon of Hybrid Automated System for Storage Protection

Journal of Electrical Systems

Abstract: - While cloud storage seems dependable and scalable, the necessity for it is increasing every day. The intricacies involved in securing data in terms of privacy and confidentiality makes the storing and administration of user information at a remote location problematic. The cloud third party who is in charge of handling the issues concerned with data security present within the cloud systems could end up being an unreliable insider. In order to enhance the protection of data and ensure the confidentiality of information without any sort of interruption from cloud third parties, the current research provides an automated hybrid Cryptographic algorithm for Cloud Data Storage. Improved automated Asynchronous Encryption for Data Streams (AEDS) and enhanced automated Concurrent Encryption for Data Blocks (CEDB) are two encryption techniques that are used to achieve great performance and efficiency. We introduce a unique encryption method in Enhanced Automated Concurrent Encryption by transforming the fixed S-box in the conventional AES to a varying S-box utilising a Ternary Galois field and Complementary Ternary Galois field in the proposed work. Additionally, the Ternary Galois field and Complementary Ternary Galois field are employed to transform the matrix used for the AES mix column technique into a dynamic matrix.

Keywords: Cryptography, Advanced Encryption Standard, Elliptic Curve Cryptography, Ternary Galois Field.

I. INTRODUCTION

Swift scaling, lower costs, great reliability are the primary benefits offered by cloud computing technology and hence a popular environment for many users [1]. Consumption of the on-demand, online services such as storage, Servers, development platforms, networks, and applications in a flexible and adaptable environment is known as cloud [2]. These online services are payment based depending on the utility of the infrastructure.

Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) are the three basic classifications of service models offered by cloud computing. IaaS makes the infrastructure available to cloud customers as virtual computation machines (such as Memory, CPU, network and storage) [3]. Another type of cloud computing service is called PaaS, which provides a platform for consumers to develop, deploy, and maintain applications devoid of getting into the specifics of how their infrastructure was built [4]. Last but not least, SaaS is a unique type of business service delivery methods offered by cloud computing that restricts customer usage to the specified apps [5]. Unauthorised access, unauthorised alteration, and loss of information integrity are just a few examples of the threats that data sources are more susceptible to when stored in cloud data centres, which are the primary limitations in a cloud platform [4, 6]. As a result, the storage of data in a cloud environment involves a variety of security concerns, including confidentiality, integrity and availability [7]. Making data available means enabling authorised people to access and utilise it whenever they need to [8]. The methodology of guaranteeing data integrity is to ensure that information accurateness and reliability, is not lost or altered by any intruders [9].

In cloud contexts, where the service providers in the cloud platform should secure authenticated permission for the critical and personal data housed in the cloud. Hence data confidentiality is yet another crucial type of challenge that is often encountered. When the Service Provider utilized the assistance from the cloud third party for data security, the risk became more evident. The service provider for the cloud platform often assigns the third party a

[1] Associate Professor, Department of Electronics and Telecommunication Engineering, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India. srividyabv@gmail.com

[2] Associate Professor, Department of Electronics and Telecommunication Engineering, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India. smitha-tce@dayanandasagar.edu

[3] Assistant Professor, Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. Soumya.s@manipal.edu

[4] Associate Professor, Department of Civil Engineering, Dayananda Sagar Academy of Technology and Management, Bengaluru, Karnataka, India. harikeerthan@gmail.com

* Corresponding Author Email: srividyabv@gmail.com

number of important functions for handling data. The parameters such as Integrity of data, its analysis, auditing, monitoring and controlling the storage, outsourcing data storage to third parties, getting replies from Service Providers, validating and directing reports to users are just a few of its responsibilities [10,11,12]. Even while the Third Party performs delicate tasks, the cloud environment may still make it vulnerable. To be more precise, the third party is typically a person, who may rarely be an untrusted or a non-neutral individual who can create a threat being an insider and who is malevolent to the cloud community [13]. Several factors, including patch management, vulnerability, encryption and decryption, threat-aware identity and access management, techniques [14] can pose a danger to data confidentiality. Because of this, a substantial portion of cloud providers have a tendency to secure the data of the user to ensure sufficient confidentially outside of the third party by enciphering the information prior to uploading the data on the cloud services.

In order to address the aforementioned problems with information confidentiality in cloud computation system, this proposal intends to focus on the creation of an automated information cryptographic system based on a novel hybrid encryption/decryption algorithm that combines modified AES and modified Elliptic Curve cryptography. The data needs to be encrypted prior to exporting it on the cloud platform. The following are the proposed module's primary contributions:

The following are the significant contributions of the specified module:

- Limiting CTP operations and improving data privacy.
- Create blocks and streams of original data.
- Creating encryption keys and their mathematically related counterparts, the decryption keys, in an amount proportional to the quantity of data streams and blocks.
- In order to be used in the decryption process, the shared key and a pair of public-private keys are saved in a certified database.
- Using a modified AES algorithm over a Ternary Galois field to encrypt data blocks. Here, dynamic S-box and dynamic matrix are used during Encryption.
- Using a modified ECC technique over a Ternary Galois field to encrypt the data stream.
- Comparing the suggested methods with existing cutting-edge encryption techniques, such as AES and ECC, over the Binary Galois field.

II. Literature Review

Several recent research on cloud security have focused on removing the inclusion of the cloud third party or lowering the severe threats due to their involvement. A way to assure that the integrity of information is kept secured on a remote cloud server is with the support of a dependable third-party auditor. This technique was suggested by the authors of the paper entitled "Integrity Checking using Third Party Auditors in Cloud Storage". In the data auditing process, the auditor has been established to be an expert agent. He is given with a task of recovering a file tag, verifying the signature, and reporting in case of any invalidity. [15].

The authors S. More and S. Chaudhari have suggested a novel approach for protecting the confidentiality and integrity of data [16]. This strategy makes use of third-party cloud computing and user-based data encryption. The owner of the information initially permits certain operations on it, such as dividing it into blocks of data, encrypting and generating a hash value, subsequently concatenating, affixing a signature. This is eventually uploaded on to the remote storage.

The accountability of the Cloud Third Party is later seen in the auditing of public data. The Cloud Third Party verifies the accurateness of the data by creating hash values for each signature that is concatenated and created in the cloud environment, as well as for the blocks of encrypted cipher that the sender has obtained. Finally, the third party compares both the signatures, just to ensure that the encrypted data has not been altered. This approach has encountered multiple Cloud Third Party issues, and it significantly increases customer workload. K. M. Akhil et al. offered another approach to restrict the function of the third party [17]. They have created a technology that offers secured data without any third party intrusion. Sending encrypted data to the cloud server for storage. Encryption is achieved using the AES encryption method. With this method, the system's internal encryption and decryption are hidden from the third-party auditor.

Additionally, P. Sivakumar et al. used the AES method to secure data on the Heroku cloud [18]. Encryption and subsequent transferring of data from the use of AES demand strong security as the system uses the concept of shared key for encryption and subsequent decryption of the same. This problem might make it simple to access the original

information to retrieve it. According to A. Orobosade et al., there is yet another method for presenting a security system that is independent of a third party [19]. They have suggested a hybrid encryption technique that combines cryptographic methods that are shared key based and public-private key based that developed in order to protect user privacy and security in the cloud. These authors have presented a cryptographic system that involves Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC), to obtain a privacy paradigm. From the preceding, we may draw the following conclusions about the inadequacies of current methods for protecting data confidentiality in cloud environments:

- Because symmetric methods only require a single shared key for encryption and decryption, using a unique type of algorithm (non-hybrid approaches) is not sufficient to guarantee higher levels of security.
- The systems that use the manual, non-automated concepts use cloud third parties to ensure they are reliable and haven't been turned into a perverse insider. It is difficult to ensure this issue in actual practises.

In this work, the authors will discuss the topic of data confidentiality from a different angle than that used in the studies mentioned above. By creating a fully portable cryptographic system for Cloud Data Storage the proposed algorithm intends to reduce the involvement of the Cloud Third Party. The proposed algorithm is based on giving an autonomous system, control over all security operations on the data, starting with the user's upload of the data to the cloud platform, moving through encryption to make it prepared for loading, and concluding with decryption when the receiver requests to recover the data. The suggested algorithm is a hybrid cryptography framework that uses two different encryption methods to implement it such as; Advanced Encryption Standard (AES) and Elliptical Curve Cryptography (ECC) using Ternary Galois field.

III. METHODOLOGY

3.1 Advanced Encryption Standard

The AES algorithm was developed by Daemen and Rijmen as a symmetric encryption method. The basis of AES is typically a combination which has both substitution and permutation computations. Wherein a series of interconnected operations, need the replacement of certain outputs for inputs (termed as substitutions), while others demand the permutation of bits. All calculations in the AES algorithm employ bytes rather than bits. In order to process the 128 bits of plain text as a matrix, the 128 bits are divided into 16 bytes and arranged as a 4X4 matrix. The number of rounds in AES can be changed according to the key length. For 128-bit keys, it uses ten rounds; for 192-bit keys, 12 rounds; and for 256-bit keys, 14 rounds. The following steps are part of the AES encryption process:

- Byte substitution: The basis for these operations is a substitution box (S-box) made specifically for this function. The resulting data matrix is a four by four matrix.
- Row Shift: The elements of the rows of the matrix are moved towards the left, and omitted elements which is dropped are replaced on the right side of the row.
- Mix Columns: Each four-byte column is now modified using a specific mathematical function. The four bytes from the original column are replaced with four new bytes by this function, which accepts four input bytes. This results in the creation of a new matrix with 16 more bytes. In the final round, this phase is skipped.
- Add round key: The 16 bytes of the four by four matrix are now considered as 128 bits, and they are then XOR ed with the 128 bits of the round key. The output will be the cipher text if this is the final round. Otherwise, 16 bytes are created from the 128 bits, and a new cycle starts. The AES algorithm's encryption process is depicted in Figure 1.

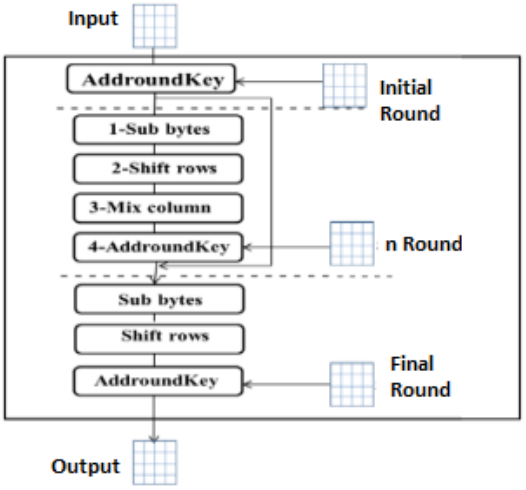

Figure 1: The Encryption Process of AES algorithm

3.2 Elliptic Curve Cryptography:

The Elliptic curve cryptography is an asymmetric cryptographic technique designed and implemented by Neil Koblitz. ECC employs dual keys for the purpose of data encryption and decryption, as is the case with any type of asymmetric method. While the private key is used solely to decrypt the data and must be kept secret, the data are encrypted using a public key that is shared with other users. The computations in the ECC algorithm can be described as follows to represent the encryption and decryption processes:-

1. The finite field considered is F_2m and P is the generator point which has the order #E (F_2m) =n
2. The Message (m_x, m_y) to be encrypted is chosen from the cubic polynomial $y^2 = x^3 + ax + b$ over F_2m
3. Key Generation:

User A chooses a unique random integer K_A from [1 to n-1] and keeps it private

Entity A computes the $A = K_A * P$ and publishes it

Similarly, User B chooses a unique random integer K_b from [1 to n-1] and keeps it private

Entity B computes the $B = K_B * P$ and publishes it

4. The Session key $(x3, y3)$ is computed using $S_{BA} = K_B \times A$
5. The Entity B computes the field element $(x4, y4)$ by considering the session key
6. Entity B generates the cipher text $Cx, Cy = \{((mx + x3) * x4), ((my + y3) * y4)\}$
7. During Decryption, entity A computes the session key $(x5, y5)$ using $S_{AB} = K_A \times B$
8. The Entity A computes the field element $(x6, y6)$ by considering the session key
9. A recovers the message $Mx, My = \{((Cx + x5) * x6), ((Cy + y5) * y6\}$
10. The Enhanced Advanced Encryption Standard and Elliptic Curve Cryptographic algorithms have been employed in this investigation because of their efficiency. As an effective encryption technique for huge quantities of data, AES is used. However, despite being a stream cipher, the ECC functions as a very sophisticated and powerful algorithm used to protect data. Using a particular encryption algorithm to ensure data privacy and confidentiality in open systems like cloud platform could make it susceptible to any sort of hacking as well as privacy violations, notwithstanding the advantages of the encryption methods listed above. As a result, as will be detailed in the subsequent section, we provide unique methodologies based on the hybridization of these traditional algorithms.

3.3. Proposed Hybrid Automated Cryptosystem for Encryption/Decryption

As depicted in figure 2, the portable cryptographic system that is being presented is a hybrid framework that combines two encryption techniques to increase the strength of the system. It relies on two fundamental principles: securing user data and protecting encryption keys. The Concurrent Encryption for Data Blocks (CEDB) uses a modified AES algorithm to encrypt user input. Instead of the conventional fixed S-box employed in the original AES, we suggest a novel random, non-repeating varying S-box and varying matrix for substitution and mix column functions, respectively. These are based on Ternary Galois fields. The intention of the update is to improve AES's level of security.

The Asynchronous Encryption for Data Streams employs Elliptic curve cryptography over Ternary Galois Field and Complimentary Galois Field to enhance performance, accuracy, and security.

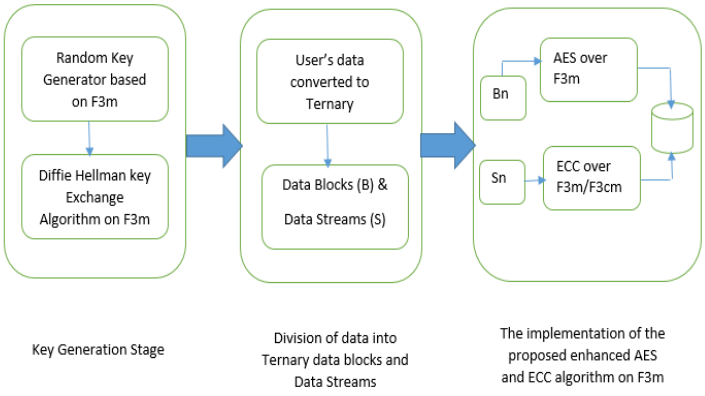

Figure 2: The Proposed Encryption/Decryption System

The suggested work starts with a Random Key Generator, which generates a random key K_i for each block of data B_i and stream of data S_i based on the Random function generator, as illustrated in Figure 3.

```
p=3;
m=input ('enter the degree of the Galois field');
primitive_polynomial=gfprimdf (m,p);
Field_elements = gftuple ([-1 :(p^m)-2]',primitive_polynomial,p);
% Field elements pertaining to F3m
key=randperm (p^m-2, 1);
```

Figure 3: Pseudo code for generating Random Keys

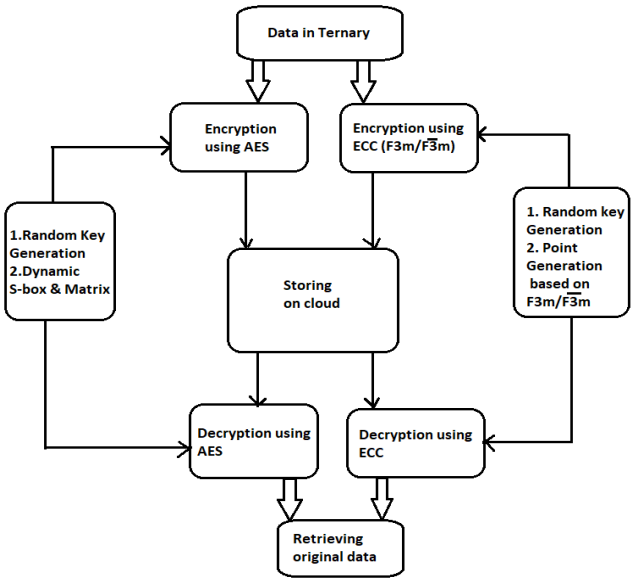

Figure 4: The Proposed Cryptographic System

As shown in figure 4, the blocks of Ternary data are encrypted and decrypted using the enhanced AES method utilising the randomly generated keys that are utilised as shared keys. Using the Diffie Hellman key exchange technique, these keys are exchanged. A unique shared key is utilised by every block of data during encryption. Additionally, the generated random keys are employed in the improved ECC algorithm's public key encryption of the data stream. The algorithm's automated elimination of the public key that is used for encryption is the core idea. These keys are kept secure and off-limits to unauthorised access by the random keys. The cloud storage is then contacted to receive all of the encrypted blocks, which are now prepared for storage.

Ternary Galois field and its 3's complement representation

This study emphasizes that the uniqueness of this research is the representation of the elements of Galois Field in its p's complement form $GF\left((\bar{p})^m\right)$ to ensure enhanced security of the information. As an illustration, let us consider GF (3^3). The elements of GF (3^3) are constructed using the primitive polynomial $p(x) = x^3 + 2x^2 + 1$. Let α be a root of the polynomial p(x).

$p(x = \alpha) = \alpha^3 + 2\alpha^2 + 1 = 0$, $\alpha^3 = -2\alpha^2 - 1$

Hence $\alpha^3 = \alpha^2 + 2$.

Table 1: The Elements of $GF(3^3)$

Elements	**Polynomial Representation**	**Ternary and Decimal representation**	**3's Complement Representation**
0	0	$(000)_3$= (0)	001
1	1	$(100)_3$=(1)	201
α	α	$(010)_3$=(3)	221
α^2	α^2	$(001)_3$=(9)	000
α^3	$2+\alpha^2$	$(201)_3$=(11)	100
α^4	$2+2\alpha+\alpha^2$	$(221)_3$=(17)	010
α^5	$2+2\alpha$	$(220)_3$=(8)	011
α^6	$2\alpha+2\alpha^2$	$(022)_3$=(24)	202
α^7	$1+\alpha^2$	$(101)_3$=(10)	200
α^8	$2+\alpha+\alpha^2$	$(211)_3$=(14)	020
α^9	$2+2\alpha+2\alpha^2$	$(222)_3$=(26)	002
α^{10}	$1+2\alpha+\alpha^2$	$(121)_3$=(16)	110
α^{11}	$2+\alpha$	$(210)_3$=(5)	021
α^{12}	$2\alpha+\alpha^2$	$(021)_3$=(15)	210
α^{13}	2	$(200)_3$=(2)	101
α^{14}	2α	$(020)_3$=(6)	211
α^{15}	$2\alpha^2$	$(002)_3$=(18)	222
α^{16}	$1+2\alpha^2$	$(102)_3$=(19)	122
α^{17}	$1+\alpha+2\alpha^2$	$(112)_3$=(22)	112
α^{18}	$1+\alpha$	$(110)_3$=(4)	121
α^{19}	$\alpha+\alpha^2$	$(011)_3$=(12)	212
α^{20}	$2+2\alpha^2$	$(202)_3$=(20)	022
α^{21}	$1+2\alpha+2\alpha^2$	$(122)_3$=(25)	102
α^{22}	$1+\alpha+\alpha^2$	$(111)_3$=(13)	120
α^{23}	$2+\alpha+2\alpha^2$	$(212)_3$=(23)	012
α^{24}	$1+2\alpha$	$(120)_3$=(7)	111
α^{25}	$\alpha+2\alpha^2$	$(012)_3$=(21)	212
α^{26}	1	$(100)_3$=(1)	001

As can be seen, Table 1 shows the elements of $GF(3^3)$ constructed using the primitive polynomial $p(x) = x^3 + 2x^2 + 1$. The Polynomial representation, the Ternary representation, the Decimal representation and the proposed novel complimentary representation of the Galois Field elements are depicted in Table 1. This complimentary representation of the data is used for all the mathematical operations like addition, multiplication, inverse and all the other relevant modular operations in the proposed algorithm.

3.4. Enhanced Automated Concurrent Encryption for Data Blocks (CEDB)

Another innovative method for the Encryption of data suggested in this framework to improve data privacy and security is the automated random cryptography methodology. This encryption method's unique feature is encrypting the data segments using an enhanced Advanced Encryption Standard (AES). The unique feature of this algorithm is the dynamic S-box and dynamic matrix generated randomly without any repetition for performing the substitution and Mix column functions that are a part of the AES algorithm, as shown in Figure 4. During the Encryption of each data block, a dynamic random S-box and dynamic random matrix are generated for substitution and mix column operation. The inverse of the dynamic s-box and the matrix is utilized for Decryption. The Encryption steps are illustrated in Algorithm1 and decryption processes are elaborated in Algorithm 2a.

Algorithm 1a: Encryption Process

Step 1: Input Random key (referred to as Master key), convert it to Ternary over F_3m

Illustration:

Plaintext in Binary
:{ 000110,001001, 101001, 010110}

Plaintext in Ternary :{ 012,021,221,112} over $F_3 3$

Plaintext in F_3m *:{* $\alpha^{15}, \alpha^{17}, \alpha^{23}, \alpha^{14}\}$

Step1a: Generation of sub keys w0, w1, w2, w3, w4 and w5 from the Master Key
The sub-keys w0 and w1 are derived from the Master Key
The sub-key w0 (1, 2) is the two left-most symbols of the master key
The sub-key w1 (1, 2) is the two right-most symbols of the master key
The other sub-keys are generated as follows:

$$w2(1,2) = \left\{(s(w1(1)), x^{i+2}) \oplus (s(w1(2)), \alpha^2)\right\},$$

Where i=1, for round 1 and α^2 is chosen for computing w2

$$w3(1,2) = \{(s(w0(1)), x^{i+2}) \oplus (s(w0(2)), \alpha^3)\},$$

Where i=1, for round 1 and α^3 is chosen for computing w3

$$w4(1,2) = \left\{(s(w2(1)), x^{i+2}) \oplus (s(w2(2)), \alpha^4)\right\}$$

Where i=2, for round 2 and α^4 is chosen for computing w4

$$w5(1,2) = \left\{(s(w3(1)), x^{i+2}) \oplus (s(w3(2)), \alpha^5)\right\}$$

Where i=2, for round 2 and α^5 is chosen for computing w5

Illustration:

$$w0(1,2) = \{\alpha^{15}, \alpha^{17}\} \quad w1(1,2) = \{\alpha^{23}, \alpha^{14}\}$$

To compute w2, w3, w4 and w5, a dynamic S-box is defined. As an illustration, dynamic s-box over F_3m is defined as follows:

Table 2: S box using Ternary Galois Field

F_3m	$(00)_3$	$(01)_3$	$(02)_3$	$(10)_3$	$(11)_3$	$(12)_3$	$(20)_3$	$(21)_3$	$(22)_3$
$(0)_3$	$\alpha^{-\inf}$	α^{23}	α^{12}	α^{17}	α^{6}	α^{9}	α^{1}	α^{4}	α^{18}
$(1)_3$	α^{5}	α^{10}	α^{11}	α^{2}	α^{16}	α^{14}	α^{22}	α^{20}	α^{8}
$(2)_3$	α^{19}	α^{15}	α^{13}	α^{24}	α^{3}	α^{7}	α^{25}	α^{21}	α^{0}

$$w2(1,2) = \{((s(w1(1)), x^{i+2}) \oplus ((s(w1(2)), \alpha^2)\},$$
$$w2(1,2) = \{((s(\alpha^{23}), \alpha^3) \oplus ((s(\alpha^{14}), \alpha^2)\}$$

Determining $s(\alpha^{23})$ *:*

Consider the degree of the finite element whose substitution value needs to be determined. The degree of the element is 23, which is equal to $(212)_3$. The value at the intersection of the row 2 and column 12 is α^7

Hence $w2(1,2) = (\alpha^7, \alpha^3) \oplus (\alpha^{14}, \alpha^2) = \{\alpha^{15}, \alpha^{12}\}$

Similarly $w3(1,2) = \alpha^{25}, \alpha^{17}$

$$w4(1,2) = (\alpha^{15,}\alpha^9)$$
$$w5(1,2) = (\alpha, \alpha^7)$$

Step2: Input Plaintext and convert it to Ternary over F_3m

Illustration: Plaintext in F_3m *: PT= {* $\alpha^3, \alpha^{25}, \alpha^{18}, \alpha^6\}$

Step3: Encryption

Step3a: $E1 = Add(PT, Keys(w0, w1)) over F_3m$

Illustration: $E1 = (\alpha^2\alpha^{25}\alpha^{18}\alpha^6) \oplus (\alpha^{15}\alpha^{17}\alpha^{23}\alpha^{14}) = (\alpha^{13}\alpha^{10}\alpha^6\alpha^{25})$

Step3b: E2=S (E1)

Illustration: $E2 = \{S(\alpha^{13})S(\alpha^{10})S(\alpha^6)S(\alpha^{25})\} = \{\alpha^{16}\alpha^{10}\alpha^1\alpha^{21}\}$

Step3c: Swapping the 2nd and 4th position ternary field element of E2

Illustration: $E3 = \{\alpha^{16}\alpha^{21}\alpha^1\alpha^{10}\}$

Step3d: E4=Mix Column

In this step, a dynamic F_3m matrix is generated, and matrix multiplication is performed with the results of E3.

$$E4 = \begin{bmatrix} \alpha^2 & \alpha^4 \\ \alpha^4 & \alpha^2 \end{bmatrix} \begin{bmatrix} \alpha^{16} & \alpha^1 \\ \alpha^{21} & \alpha^{10} \end{bmatrix} = \begin{bmatrix} \alpha^{21} & \alpha^4 \\ \alpha^{22} & \alpha^8 \end{bmatrix}$$

Step3e: $E5 = Add(E4, sub\_keys(w2, w3))$

Step3f: E6=S (E5)

Step3g: E7=Swapping the 2nd and 4th position ternary field element of E2

Step3h: $CT = E8 = Add(E7, sub\_keys(w4, w5))$

After step3h, the Cipher text obtained is $\{\alpha^{24}\alpha^{23}\alpha^6\alpha^{20}\}$

Algorithm 2a: Decryption Process

Step4: Decryption

Step 4a: $D1 = Subtract(CT, Keys(w4, w5))overF_3m$

$$D1 = Subtract([\alpha^{24}\alpha^{23}\alpha^{6}\alpha^{20}],[\alpha^{15}\alpha^{9}\alpha\alpha^{7}])overF_3m$$

Illustration: $= (\alpha^{10}\alpha^{14}\alpha^{25}\alpha^{7})$

Step 4b: D2=Swapping the 2nd and 4th position ternary field element of D1

Illustration $D2 = (\alpha^{10}\alpha^{7}\alpha^{25}\alpha^{14})$

Step 4c: Inverse S box substitution D3=IS (D2)

During the encryption process, the S-box is generated randomly. The Inverse S box is computed as soon as the dynamic S-box is randomly generated using the mathematical properties related to F_3m. The Inverse S box IS.

Table 3: Inverse S-box using Ternary Galois Field

F_3m	$(00)_3$	$(01)_3$	$(02)_3$	$(10)_3$	$(11)_3$	$(12)_3$	$(20)_3$	$(21)_3$	$(22)_3$
$(0)_3$	$\alpha^{-\inf}$	α^{6}	α^{12}	α^{22}	α^{7}	α^{9}	α^{4}	α^{23}	α^{17}
$(1)_3$	α^{5}	α^{10}	α^{11}	α^{2}	α^{20}	α^{14}	α^{19}	α^{13}	α^{3}
$(2)_3$	α^{8}	α^{18}	α^{16}	α^{25}	α^{15}	α^{1}	α^{21}	α^{24}	α^{0}

Illustration D3=Inverse S box Substitution= IS (D2)

$$D3 = (\alpha^{10}\alpha^{23}\alpha^{24}\alpha^{14})$$

Step 4d: $D4 = Subtract(D3, Keys(w2, w3))overF_3m$

Illustration $D4 = (\alpha^{21}\alpha^{4}\alpha^{22}\alpha^{8})_{=}\begin{bmatrix}\alpha^{21} & \alpha^{22}\\ \alpha^{4} & \alpha^{8}\end{bmatrix}$

Step 4e: D5= Inverse Mix column;

Matrix Multiplication of the inverse of the random matrix generated during Encryption and the results D4

$$D5 = D4 \times \begin{bmatrix}\alpha & \alpha^{16}\\ \alpha^{16} & \alpha\end{bmatrix} overF_3m$$

$$D5 = \begin{bmatrix}\alpha^{21} & \alpha^{22}\\ \alpha^{4} & \alpha^{8}\end{bmatrix} \times \begin{bmatrix}\alpha & \alpha^{16}\\ \alpha^{16} & \alpha\end{bmatrix} = \begin{bmatrix}\alpha^{16} & \alpha\\ \alpha^{21} & \alpha^{10}\end{bmatrix}$$

$$= [\alpha^{16}, \alpha^{21}, \alpha, \alpha^{10}]$$

Step 4f: D6= Swapping the 2nd and 4th Ternary field digit of D5

Illustration $D6 = [\alpha^{16}, \alpha^{10}, \alpha, \alpha^{21}]$

Step 4g: D7=Inverse S box Substitution= IS (D6)

Illustration $D7 = [\alpha^{13}, \alpha^{10}, \alpha^{6}, \alpha^{25}]$

Step 4h: The recovered Plain text, $D8 = Subtract(D7, Keys(w0, w1))overF_3m$

Illustration: Recovered Plaintext in F_3m *: D8= {* $\alpha^{3}, \alpha^{25}, \alpha^{18}, \alpha^{6}\}$

Note: The values considered for illustration purpose is F_33 **. Higher values of 'm' are considered for implementation.**

3.5. Improved automated Asynchronous Encryption for Data Streams (AEDS)

The improved automated asynchronous Encryption for data streams uses Elliptic Curve cryptography over F_3m and the complimentary Ternary Galois field F_3m. The algorithm for Encryption and Decryption using the Ternary Galois field and the Complimentary Ternary Galois field is described subsequently. Algorithm 1b, 2b illustrates the encryption and decryption process using the Ternary Galois field, while Algorithm 1c, 2c illustrates the methodology for encrypting and decrypting the data using Complimentary Ternary Galois field:

Algorithm 1b: Encryption using F_3m
ECC over finite field $GF(3^3)$

Consider the finite field F_3 m, and the generator point $p = \{\alpha^3, \alpha^5\}$ and having the order E (F_3 m) = n = 27;

Step 1: -Key generation

User A carries out the following operation

A chooses a unique random integer K_A = 3 between $1 \prec K_A \leq n-1$

User A computes the point denoted as A = $KA * P$ =3 $* (\alpha^3, \alpha^5) = \alpha^{12}, \alpha^{18}$

Step 2: Encryption process

Entity B Sends Message to A

Message M = (10100000100) = {$(220)_3$, $(020)_3$)} = (α^5, α^{14})

User B carries out the following operations:

- Chooses the public key of A from the directory of public keys: A = α^{12}, α^{18}

The Plaintext M is represented as a pair of points on the x-axis and y-axis($M1, M2$)

M1 = (101000), (00100) = (220), (020) over$GF(3^m)$

- Selects the random integer K_B = 2 from [1, n-1]
- Computes the public key: B = $KB\ P$=2 $* (\alpha^3, \alpha^5) = (\alpha^{18}, \alpha^{24})$
- Computes the shared Key = $SBA = KB * A$=2 $* (\alpha^{12}, \alpha^{18})$

$(x3, y3) = (\alpha^{20}, \alpha^{22})$

The shared key SBA is treated as the Session Key.

- Computes the field element by considering the session key and raising it to the order of the Galois field m

$(x4, y4) = ((SBA(x))^M, ((SBA(y))^M$

$(x4, y4)$=$((\alpha^{20})^3, (\alpha^{22})^3)$ over $GF(3^3)$

=(α^8, α^{14})

- B forms the cipher text

$c1 = ((m1 + x3)(x4))$=α^3

$C2 = (m2 + y3)(y4)$=α^{21}

B transmits the public key of A and the cipher text to A; $(\alpha^{12}, \alpha^{18}, \alpha^3, \alpha^{21})$

Algorithm 2b: Decryption using F_3m

Step 3: Decryption process (A)

Entity A decrypts the cipher $(\alpha^{12}, \alpha^{18}, \alpha^3, \alpha^{21})$ received from the user B by performing the following steps:

A computes the session key

$SAB = KA(B) = (2)(\alpha^{18}, \alpha^{24})$

$(x3, y3) = (\alpha^{20}, \alpha^{22})$

A forms $(x4, y4)$ just as B did $x4 = (\alpha^{(20)^3})$=α^5

$y4 = (\alpha^{(22)^3})$=α^{14}

A recovers the message (M1, M2) by Computing

M1 = $\frac{C1}{x4}$ − x3 over$GF(3^m)$

= $\frac{\alpha^3}{\alpha^8} - \alpha^{20}$=$\alpha^5$

M2 = $\frac{C2}{y4}$ − $y3$ over$GF(3^m)$

$= \frac{\alpha^{21}}{\alpha^{14}} - \alpha^{22} = \alpha^{14}$

(M1, M2) = (α^5, α^{14}) = (220,020) = (101000001000)

ECC over finite field $GF(\bar{3}^m)$

The finite field considered is $F_{\bar{3}}m$ and the generator point $p = \{\alpha^0, \alpha^{19}\}$ having the order E ($F_{\bar{3}}m$) = n = 27;

Algorithm 1c: Encryption using $F_{\bar{3}}m$

Step 1:-Key generation

Entity A performs the following operation

A selects a random integer K_A= 3 from [1, n-1]

A computes the point A=$KA\ P$=$3 * \{\alpha^0, \alpha^{19}\} = (\alpha^5, \alpha^7)$

Step 2: Encryption process

Entity B perform the following steps.

Looks up at the public key of A from the public key server: A=(α^5, α^7)

Entity B Sends Message to A

Message M= (10100000100)

Represent M as a pair of elements $(M1, M2)$

M1= (101000), (00100) = (220), (020) over $GF(\bar{3}^m)$

=(α^{19}, α^8) over$GF(\bar{3}^m)$

Selects the random integer K_B = 2 from [1, n-1]

Computes the public key: B = $KB\ P$=$2 * (\alpha^0, \alpha^{19})$

= $(\alpha^{17}, \alpha^{17})$

Computes the shared Key =$SBA = KB * A$=$2 * (\alpha^5, \alpha^7)$

$(x3, y3) = (\alpha^{14}, \alpha^4)$

The shared key SBA is treated as the Session Key.

Computes the field element by considering the session key and raising it to the order of the Galois field m.

$(x4, y4) = ((SBA(x))^M, ((SBA(y))^M$

$(x4, y4)$=$((\alpha^{(14)^3}, (\alpha^{(4)}$ over $GF(3^3)$

=$(\alpha^{16}, \alpha^{12})$

B foams the cipher text

$c1 = ((m1 + x3)(x4))$=α^{18}

$C2 = (m2 + y3)(y4)$=α^4

Entity B transmits the public key of A and the cipher text to A; $(\alpha^5, \alpha^7, \alpha^{18}, \alpha^2)$

Algorithm 2c: Decryption using $F_{\bar{3}}m$

Step 3: Decryption process (A)

Entity A decrypts the cipher $(\alpha^5, \alpha^7, \alpha^{18}, \alpha^2)$ obtained from user B after performing the following computations:

A performs the following computation to obtain the session

$SAB = KA(B) = (2)(\alpha^{18}, \alpha^{24})$

$(x3, y3) = (\alpha^{14}, \alpha^4)$

A form $(x4, y4)$ just as B did $x4 = (\alpha^{14})^3$ =α^{16}

$y4 = (\alpha^4)^3$=α^{12}

A recovers the message (M1, M2) using the following equations:

M1=$\frac{C1}{x4} - x3$ over$GF(\bar{3}^m)$

= $\frac{\alpha^{18}}{\alpha^{16}} - \alpha^{14}$=$\alpha^{19}$

M2=$\frac{C2}{y4}$–y3 over$GF(3^{\bar{m}})$ =$\frac{\alpha^2}{\alpha^{12}} - \alpha^4$=$\alpha^8$

(M1, M2) = (α^{19}, α^{8}) over$GF(3^{\bar{m}})$= (220,020) $GF(\bar{3}^{m})$ = (101000001000)
In this approach, point addition over$GF(\bar{3}^{m})$has been used.

IV. Results and Discussion

The proposed hybrid cryptographic system has been designed, implemented and tested. The implementation has been performed using MATLAB 2023.

Randomness Test

Runstest returns a test decision for the null hypothesis that the values in the data vector x come in random order against the alternative that they do not. The Runstest returns two parameters, 'h' and 'p'. The returned value of h = 0 indicates that the runs test does not reject the null hypothesis that the values in x are in random order at the default 5% significance level. The p-value of the test returns as a scalar value, which is in the range [0, 1]. 'p' is the probability of observing a test statistic as extreme as, or more extreme than, the observed value under the null hypothesis. Small values of p cast doubt on the validity of the null hypothesis. The generation of the input data, substitution box (S-box), and the matrix in the mix column operation is tested for randomness. Similarly, the randomness of the cipher text concerning the plaintext is also tested. The randomness test is employed for the Enhanced AES over Binary Galois field, Enhanced AES over Ternary Galois field, and Enhanced ECC over Binary Galois field, Enhanced ECC over Ternary Galois field and the proposed hybrid cryptosystem over Ternary Galois field.

Illustration:

N=length of the input sequence
For i=1: N
[h (i), p (i)]=runstest (c (i), m (i));
End
/ C is the cipher text */*
/ m is the plain text */*
The experimental results obtained yield a value of h=0 and p=1 for all the test cases. This shows the randomness in the input data generation, randomness in the S-box, randomness in the matrix and randomness in the computed cipher text. If the data pattern and the cipher text cannot be predicted, the brute force attack, a plaintext-cipher text pair attack, is not possible. The security offered is enhanced.

4.1. Performance Metrics

The proposed encryption methods that work with ternary data are contrasted with those that use binary data. Different sizes of Ternary data are used to implement and test the experiments. Each file is broken up into blocks that consist of 128MB. The second block of 128MB is divided into smaller data streams and encrypted using enhanced ECC over Ternary fields, while the first block of 128MB is encrypted using the improved AES technique over Ternary fields. The following statistical results are determined for all the algorithms used in the comparison to assess the encryption algorithms: the Encryption and Decryption of files and the throughput determine after the encryption and decryption files.

Simulation time taken for encryption/decryption of data: The total amount of time taken by the system to perform Encryption or Decryption of all the data blocks of a file that is being selected using a particular algorithm from the commencement of encryption or decryption process till its conclusion.

$$T_S = \sum((End\_time) - (start\_time)), perblock$$

Processing time (PT): It is the total simulation time taken by the system for random key generation and encrypting or decrypting process. It is computed using the equation.

$$PT = keygeneration\_time + T_S$$

Throughput: It is defined as the amount of data that can be processed in a unit time successfully. It is computed using the equation:
$Throughput$ (MB/s) = Size of the selected file (in MB) /T (in seconds)

4.2. Result Analysis and Discussions

The proposed technique employs unbalanced Ternary digits, denoted as 0, 1, and 2. Compared to binary data, ternary data could fit more volumes of data in the same available storage. The comparison table shows how well the modified AES algorithm, enhanced ECC algorithm, and hybrid cryptosystem using binary field and ternary field perform when encrypting and decrypting various files of varying sizes. The files are divided into two blocks of equal size. These equal-sized blocks are encrypted concurrently using the proposed hybrid cryptographic algorithm. In addition, the Ternary field-based hybrid cryptosystem eliminates the third-party problem and achieves high security. The results are tabulated after implementing Modified AES (using Binary and Ternary Galois field), Modified ECC (using Binary, Ternary, and Complimentary Ternary Galois field) and the proposed hybrid algorithm over the Ternary Galois field. The tabulation displays the total amount of time used to encrypt and decrypt data blocks of various sizes, including the key generation time. The time taken for generating a random s-box, its inverse, random matrix and its inverse are also considered as a part of the key generation time.

Table 4. Comparison between the proposed algorithm and the other tested algorithms

File Size	**Overall Encryption Time**					**Overall Decryption Time**				
	AES (Binary)	**ECC (Binary)**	**AES (Ternary)**	**ECC (Ternary)**	**Proposed Cryptosystem (Ternary)**	**AES (Binary)**	**ECC (Binary)**	**AES (Ternary)**	**ECC (Ternary)**	**Proposed Cryptosystem (Ternary)**
98KB	2.8703	2.619	2.5394	2.2739	1.8381	28.1516	39.8512	26.6206	39.0814	18.3224
196KB	6.2803	5.7195	5.2281	4.0962	2.6087	56.7898	78.9381	53.5651	72.8808	28.1987
368KB	11.9245	10.8263	9.8533	8.1337	5.1605	85.5689	110.4939	74.9554	102.0143	56.3405
736KB	14.3076	12.9821	11.8706	9.7455	8.2016	103.0231	132.5833	89.9287	122.8972	67.8713
3MB	48.9596	40.2215	41.7146	33.5358	29.9334	488.1295	590.822	456.7043	578.3877	478.1881
6MB	57.1572	48.2564	51.2228	40.2315	35.9087	586.2389	709.8041	548.7122	694.0559	574.5189
128MB	191.2465	58.7076	167.2428	50.3156	85.8111	910.9861	972.6982	875.4947	901.2891	859.5789
256MB	301.649	180.1892	267.535	100.5743	170.2163	1017.15	1120.937	954.3208	1081.537	903.5737

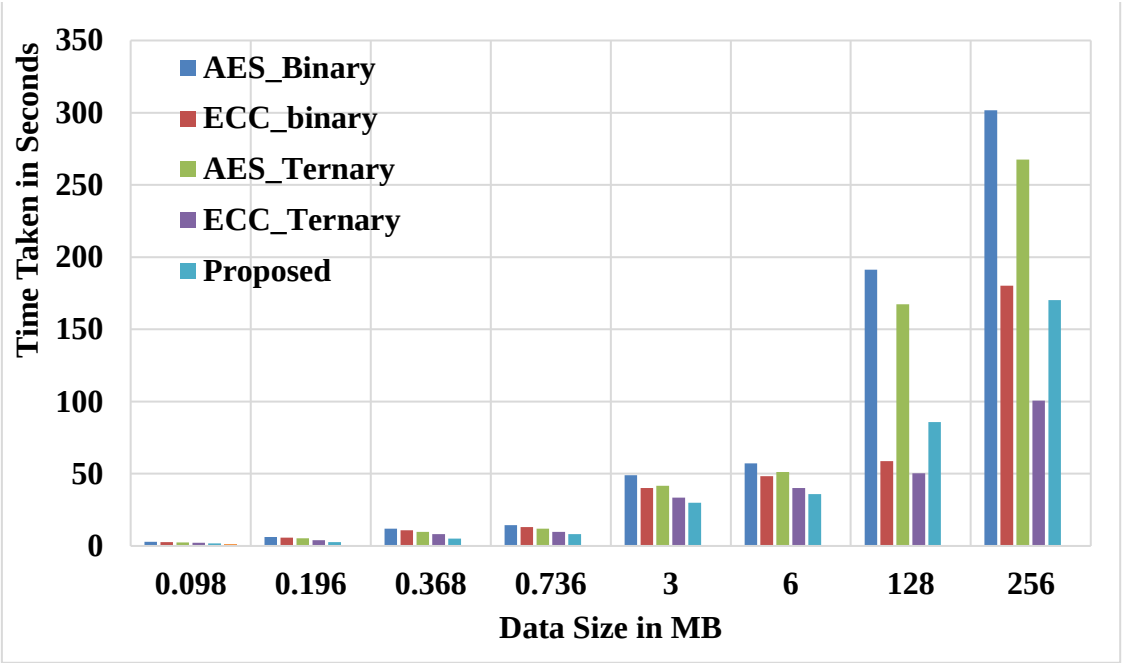

Figure 5: Overall Encryption time taken by the various tested algorithms and the proposed algorithm

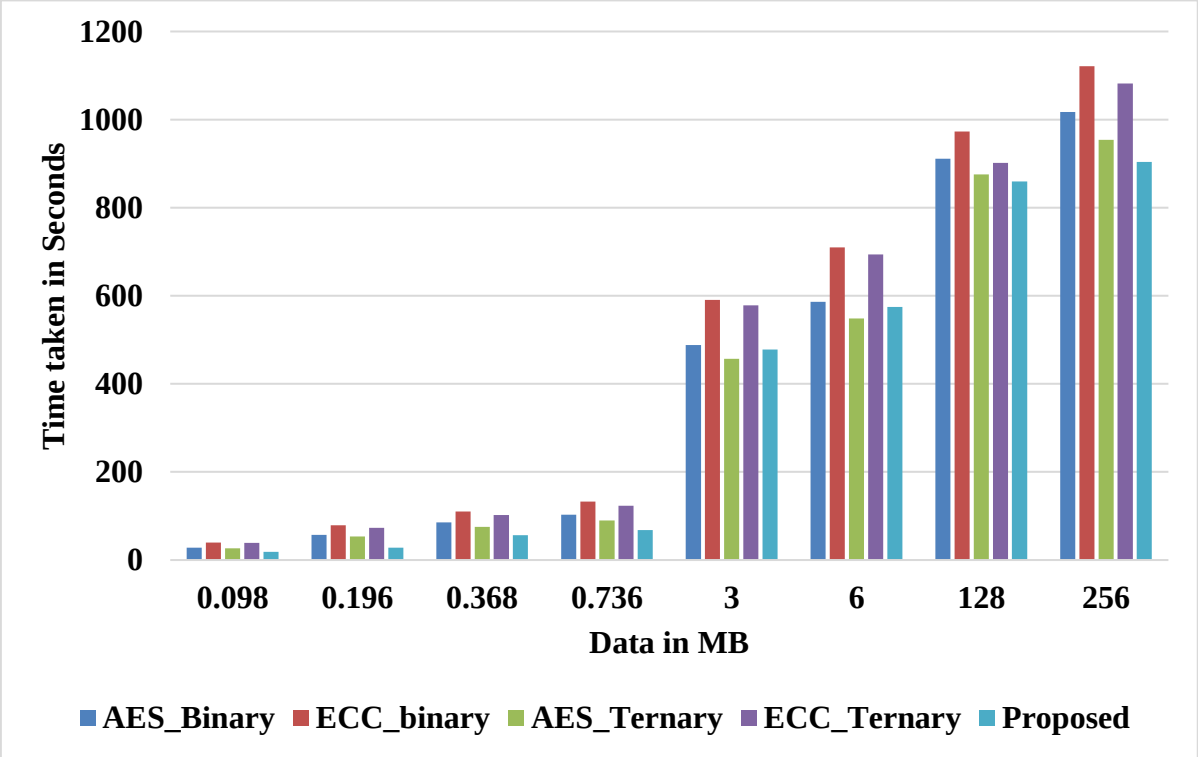

Figure 6: Overall Decryption time taken by the various tested algorithm and the proposed algorithm.

According to Table 4, Figure 5, it can be observed that the overall time taken for Random key generation, random S-box generation, Random matrix generation and Encryption using the proposed hybrid cryptographic algorithm is lesser when compared to the other tested algorithms. Also all the computations using Ternary modular Galois field consumes lesser time compared to Binary modular Galois field. As per the results tabulated and figure 6, it can be observed that the overall time taken for Decryption using the proposed technique is lesser when compared to the other tested algorithms. This overall time includes the inverse s-box computation and inverse matrix generation using Ternary Galois field.

Table 5: Comparison between the proposed algorithm and the other tested algorithms

File Size	Time for Encryption					Time for Decryption				
	AES (Binary)	ECC (Binary)	Enhanced AES (Ternary)	Enhanced ECC (Ternary)	Proposed Cryptosystem (Ternary)	AES (Binary)	ECC (Binary)	Enhanced AES (Ternary)	Enhanced ECC (Ternary)	Proposed Cryptosystem (Ternary)
	Executed Individually	Executed Individually	Executed Individually	Executed Individually	Enhanced AES and Enhanced ECC are execute Concurrently	Executed Individually	Executed Individually	Executed Individually	Executed Individually	Enhanced AES and Enhanced ECC are execute Concurrently
98KB	2.8614	2.5721	2.4504	2.2170	1.7812	28.1427	39.8043	26.5316	39.0345	18.2655
196 KB	6.2714	5.6726	5.1391	4.0393`	2.5518	56.7809	78.8912	53.4761	72.8339	28.1418
386 KB	11.9156	10.7794	9.7643	8.0768	5.1036	85.56	110.447	74.8664	101.9674	56.2836
736 KB	14.2987	12.9352	11.7816	9.6886	8.1447	103.0142	132.5364	89.8397	122.8503	67.8144
3MB	48.9507	40.1746	41.6256	33.4789	29.8765	488.1206	590.7751	456.6153	578.3408	478.1312
6MB	57.1483	48.2095	51.1338	40.1746	35.8518	586.23	709.7572	548.6232	694.00896	574.462
128 MB	191.2376	58.6607	167.1538	50.2587	85.7542	910.9772	972.6513	875.4057	901.2422	859.522
256 MB	301.6401	180.1423	267.4460	100.5174	170.1594	1017.1413	1120.89	954.2318	1081.49	903.5168

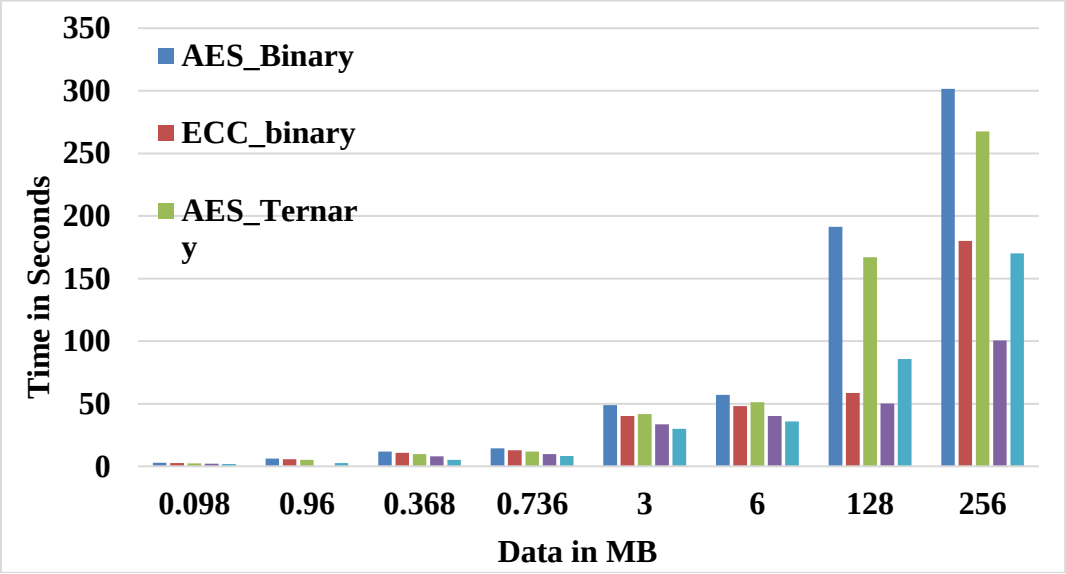

Figure 7: Time taken for Encryption by the Tested algorithms and the proposed algorithm

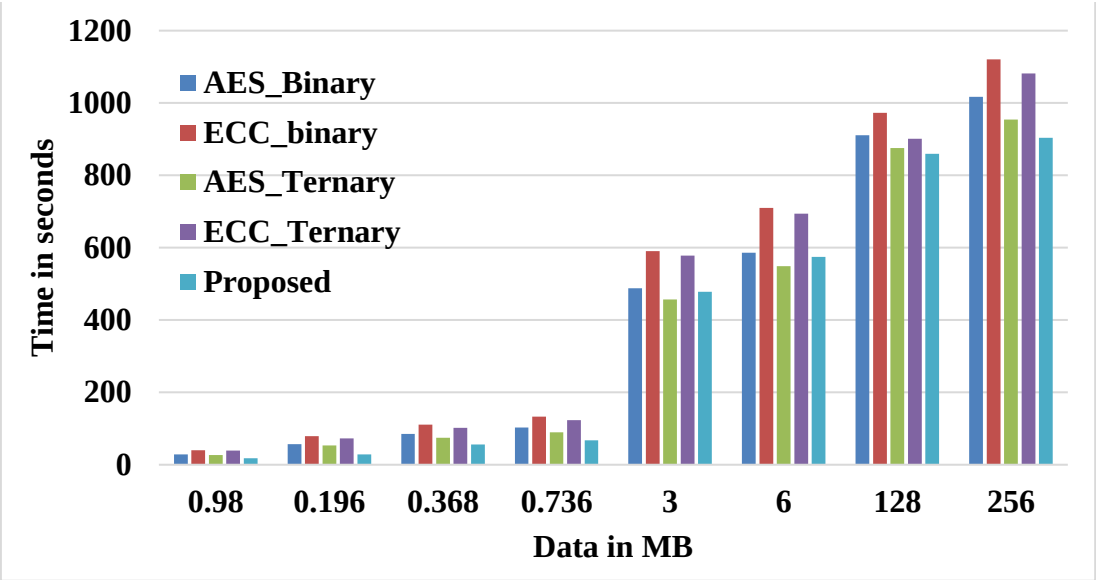

Figure 8: Time taken for Encryption by the Tested algorithms and the proposed algorithm

Figure 7 and 8 shows the time taken to encrypt and decrypt data of various sizes using the tested algorithms such as Modified AES (using Binary and Ternary Galois field), Modified ECC (using Binary, Ternary, Complimentary Ternary Galois field) and the proposed hybrid algorithm over Ternary Galois field. It is observed that the proposed algorithm has a lesser encryption and decryption time as the data is processed concurrently. From Table 4 and 5, it can be observed that the proposed hybrid encryption/decryption technique executes at a faster rate, compared to the Enhanced AES technique over Binary and Ternary Galois field as well as the Enhanced ECC technique over Binary and Ternary Galois field. The reason is due to the fact, that data is divided into blocks and each block is concurrently executed by the symmetric key based enhanced AES and asymmetric key based enhanced ECC algorithms. Further it can be observed that the representation of data in Ternary and computations in Ternary enhances the speed of execution compared to data in Binary. In contrast, the Ternary data-based hybrid cryptosystem took less time to process big file sizes. These methods outperformed the competition in terms of time savings despite having more difficult operations during the decryption process and taking longer than during Encryption. They guarantee improved security. Additionally, the hybrid cryptosystem using Ternary field and the upgraded AES encryption/decryption process both use dynamic S boxes and dynamic matrices for each every data block, increasing security.

V. Conclusion and Future Work

Users are now storing the data on remote server, as a result of the growth in data volume. Data that has been outsourced is no longer under the control of the user and is open to unauthorized, unethical actions. In order to ensure data security and lessen the effort on users to protect their own data, we have built an automated hybrid cryptographic system for this proposed method. This system operates without intervention from cloud third parties. The hybrid cryptographic system is more secure since it employs Ternary field data instead of traditional binary data. Furthermore, to increase security the hybrid cryptographic system uses a random dynamic S-box and dynamic matrix for the substitution and mix column techniques for each data block. Also, to reduce the overall processing time, data is split into blocks of equal size and encrypted using Enhanced AES over Ternary Galois field and Enhanced ECC over Ternary Galois Field by using concurrent processing. Since the data is represented in Ternary, more amount of information can be stored in the space allocated.

As a scope of further improvement, the algorithms that provide information security can be implement on any embedded platform.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

"Conceptualization, Srividya B V and Smitha Sasi; Methodology, Srividya B V; software, Smitha Sasi; validation, Srividya B V, Smitha Sasi, and Prem kumar; formal analysis, Srividya B V; investigation, Smitha Sasi; resources, Srividya B V; data curation, Smitha Sasi; writing—original draft preparation, Srividya B V; writing—review and editing, Prem Kumar; visualization, Srividya B V; supervision, Srividya B V; project administration, Srividya B V;

REFERENCES

[1] Noha E El-Attar, Doaa S El-Morshedy, Wael A Awad, "A New Hybrid Automated Security Framework to Cloud Storage System", Journal of Cryptography, MDPI, 2021

[2] Dhanapal, R.; Tharageswari, K.; Karthik, S. A decentralized accountability framework for enhancing secure data sharing through ICM in cloud. Innov. Technol. Explor. Eng. 2019, 8, 1505–1511, ISSN 2278-3075.

[3] Shahzadi, S.; Iqbal, M.; Qayyum, Z.U.; Dagiuklas, T. Infrastructure as a service (IaaS): A comparative performance analysis of open-source cloud platforms. In Proceedings of the International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, Lund, Sweden, 19–21 June 2017; pp. 1–6.

[4] Jathanna, R.; Jagli, D. Cloud Computing and security issues. Int. J. Eng. Res. Appl. 2017, 7, 31–38, ISSN 2248-9622.

[5] Kulkarni, G.; Waghmare, R.; Palwe, R.; Waykule, V.; Bankar, H. Cloud storage architecture. In Proceedings of the International Conference on Telecommunication Systems, Services, and Applications, Denpasar-Bali, Indonesia, 30–31 October 2012; pp. 76–81.

[6] Vurukonda, N.; Rao, B.T. A Study on data storage security issues in cloud computing. In Proceedings of the Inernational Conference on Intelligent Computing, Communications & Convergence, Odisha, India, 24–25 January 2016; Volume 92, pp. 128–135.

[7] Bentajer, A.; Hedabou, M.; Abouelmehdi, K.; Elfezazi, S. CS-IBE: A data confidentiality system in a public cloud storage system. Procedia Comput. Sci. 2018, 141, 559–564.

[8] Tawalbeh, L.A.; Saldamli, G. Reconsidering big data security and privacy in cloud and mobile cloud systems. Comput. Inf. Sci. 2019, 33, 810–819.

[9] Das, D. Secure cloud computing algorithm using homomorphic Encryption and multi-party computation. In Proceedings of the International Conference on Information Networking, Chiang Mai, Thailand, 10–12 January 2018; Volume 1, pp. 391–396.

[10] Mohta, A.; Awasthi, L.K. Cloud data security while using third-party auditor. Sci. Eng. Res. 2012, 3, 1–9, ISSN 2229-5518.

[11] Yusop, Z.M.; Abawajy, J.H. Analysis of insiders attack mitigation strategies. Procedia-Soc. Behav. Sci. 2014, 129, 611–618.

[12] Singh, S.; Thokchom, S. Public integrity auditing for shared dynamic cloud data. In Proceedings of the International Conference on Smart Computing and Communications, Kurukshetra, India, 7–8 December 2018; Volume 125, pp. 698–708.

[13] Potey, M.M.; Dhote, C.A.; Sharma, D.H. Homomorphic Encryption for security of cloud data. In Proceedings of the International Conference on Communication, Computing and Virtualization, Mumbai, India, 26–27 February 2016; Volume 79, pp. 175–181.

[14] El-Attar, N.E.; Awad, W.A.; Omara, F.A. Empirical assessment for security risk and availability in public cloud frameworks. In Proceedings of the International Conference on Computer Engineering & Systems, Cairo, Egypt, 20–21 December 2016; pp. 17–25.

[15] Chakraborty, S.; Singh, S.; Thokchom, S. Integrity Checking using third party auditor in cloud storage. In Proceedings of the International Conference on Contemporary Computing, Noida, India, 2–4 August 2018; pp. 1–6.

[16] More, S.; Chaudhari, S. Third party public auditing scheme for cloud storage. In Proceedings of the International Conference on Communication, Computing and Virtualization, Mumbai, India, 26–27 February 2016; Volume 79, pp. 69–76.

[17] Akhil, K.M.; Kumar, M.P.; Pushpa, B.R. Enhanced cloud data security using AES algorithm. In Proceedings of the International Conference on Intelligent Computing and Control, Coimbatore, India, 14–15 June 2018; pp. 1–5.

[18] Sivakumar, P.; NandhaKumar, M.; Jayaraj, R.; Kumaran, A.S. Securing Data and Reducing the Time Traffic Using AES Encryption with Dual Cloud. In Proceedings of the International Conference on System, Computation, Automation and Networking, Pondicherry, India, 29–30 March 2019; pp. 1–5.

[19] Orobosade, A.; Favour-Bethy, T.A.; Kayode, A.B.; Gabriel, A.J. Cloud Application Security using Hybrid Encryption. Commun Appl. Electron. 2020, 7, 25–31, ISSN 2394-4714.