[1]Maria Navin J R

[2]Nagaraj M Lutimath

An Enhanced AES-ECC model with Key Dependent Dynamic S-Box for the Security of Mobile Applications using Cloud Computing

Journal of Electrical Systems

Abstract: **-** The spectacular growth of computational clouds has drawn attention and enabled intensive computing on client devices with constrained resources. Smart phones mostly use the data centre demand service model to provide data to computationally intensive applications. It is challenging to outsource sensitive and private information to distant data centres due to growing concerns about data privacy and security. Therefore, conventional security algorithms have to be upgraded to address the brand-new security concerns that have emerged in the cloud environment. The Advanced Encryption Standard (AES) algorithm encrypts and decrypts messages using the four steps, namely Sub-Bytes, Shift-Rows, Mix-Columns, and Add Round Key. It has been changed so that the Sub-byte transformation depends on the round key. Making the S box round key dependent is intended to make it such that even a little change in the key will have a big impact on the cipher text. The avalanche effect and execution time of the standard and modified AES algorithms are implemented and assessed. In order to suggest a safe and efficient solution, the study also aims to modify the shift rows and the mix column phases of AES. Using cryptographic techniques such as elliptic curve cryptography (ECC), several protocols and algorithms have been developed to guarantee the security and integrity of the data. The proposed solution combines the enhanced and updated The Advanced Encryption Standard (AES) technology with the ECC to guarantee data secrecy. The avalanche effect approach is used in this study to explore and analyze the strength and quality of the new s-box. The results demonstrated that each encryption operation generates a unique cipher text. A strong avalanche effect was also achieved with the redesigned dynamic s-box using irreducible polynomials, surpassing the strict avalanche criterion (SAC). The results show that the proposed method ECC-EAESKDS Elliptic Curve Cryptography-Enhanced AES with Key Dependent S box is efficient and yields better results than the alternatives. According to the suggested security architecture, cloud users may control data protection and integrity in a secure manner.

Keywords: Elliptic Curve Cryptography (ECC), Advanced Encryption Standard (AES), S-Box, Sub-Bytes, Shift Rows, Mix Columns, Confidentiality, Mobile Device, Avalanche Effect, Irreducible Polynomial.

I. INTRODUCTION

The most cutting-edge and practical branch is cloud computing, which makes use of the Internet to allow users access to data storage. Researchers have paid a lot of attention to cloud computing security in recent years because of the significance of the data used and the growing use of the technology, which is now used to deliver a wide range of services in numerous industries.

Data security is thus a significant problem while keeping data in clouds. One of the most crucial ways to provide data security in the cloud is through cryptographic algorithms. If the end user has an Internet connection, they can save money by storing and retrieving private data from remote storage in a cloud computing environment. The user can access the data at any time and from any location. However, the security of data sent over the cloud cannot always be ensured. Because the end user can only access the data with the help of a third party, data integrity and authentication can be compromised. In a cloud computing environment, the end user can save money by storing and retrieving private data from remote storage if they have access to the Internet. The user has unrestricted access to the data at all times from any location. However, there is no guarantee that data sent over the cloud will always be secure.

The term mobile cloud computing refers to a new paradigm in which computing power is available as an on-demand service via mobile and tablet devices. Users may lose control of their data if they decide to use cloud computing. As a result, backing up data while in transit and in the cloud is a serious issue. Applications that rely on changing technology need to take into account all possible dangers. There are a number of problems with cloud technology, including data integrity and security. This is the main motivation behind why people are wary of utilizing cloud innovation. Access control and key management are also data security factors that contribute to the full utilization

[1]* Maria Navin J R: Department of Information Science and Engineering, Sri Venkateshwara College of Engineering, Bangalore, Karnataka, India, *Corresponding author e-mail: marianavin.jr@gmail.com

[2] Nagaraj M Lutimath: Department of Computer Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore, Karnataka, India, author e-mail: nagarajlutimath@gmail.com

of cloud technologies. The primary concerns of cloud clients in portable distributed computing are information related security issues, such as information honesty, privacy, accessibility, and discernibility.

Data security is crucial in the current world. Particularly, if the data transmission network used is insecure. He can encrypt and decrypt data using the same key thanks to symmetric key cryptography. Despite having a straightforward architecture, brute-force attacks can readily break them. The entire cryptographic security can be compromised if an attacker manages to obtain the key. Asymmetric key algorithms, on the other hand, employ key pairs (one for encryption and one for decryption) and provide greater security than symmetric key algorithms but require more time to execute.

By combining the modified Advanced Encryption Standard (AES) with ECC, this research aims to create a hybrid encryption method that protects multimedia data such as photos, text files, audio files, and video files. It makes use of a type of hybrid cryptography that uses both symmetric and asymmetric keys. The use of symmetric technology, such as AES, DES, and others, to accomplish this is susceptible to attacks using brute force. As a result, symmetric/asymmetric hybrid systems are evaluated as having security levels that are equal to or higher than those of traditional systems. A 1024-bit RSA key and a 160-bit ECC key share the same level of security. It was decided to use ECC as the provider for asymmetric keys in our research, because it occupies less space than other methods. It was in use for a very long time before the inherent limitations of symmetric encryption algorithms such as DES were exploited. DES uses a 56-bit key to encrypt 64-bit plaintext and produce 64-bit cipher text. Brute force attacks are capable of breaking even the most straightforward and straightforward encryption due to the short key length.

Triple DES (3DES), which employs two keys and triple encrypts the plaintext with a 112-bit or 168-bit key, was used to get around these restrictions. However, compared to ordinary DES, the encryption process is substantially slower. It was suggested to use the AES cipher as an encryption standard to overcome the drawbacks of DES. Depending on how many rounds are used, a single key that is 128, 192, or 256 bits long (10, 12, or 14) is used. AES often involves substitution, shifting, or rows, combining column modifications, and adding a round key to all rounds but the final one in order to obtain the associated cipher-text. Although DES is faster than AES, it does not provide the same level of security, according to a performance comparison of the techniques now in use.

Cryptography [30] is classified as symmetric or asymmetric in most of the texts on the topic. Rijndael, a participant in the US National Institutes of Standard and Technological Computing [1, 2], first introduced it in 2001. It takes the place of DES. The most widely used symmetric cipher encryption method for data security is AES. Each cycle in the secret writing method contains four steps [3]. Shift rows, Sub Byte, Round Key, and Mix Column are added. The method supports successive rounds of 10, 12, and 14 and keys with a length of 128, 192, or 256 bits [4-5].

More postponement is delivered by Sub Bytes and Blend Sections. Subsequently, the AES calculation isn't utilized for the Web of Things, remote recognition organizations, and low-power gadgets like advanced cells and PDAs. Therefore, for these contexts, a low-cost and low-power symmetric data encryption algorithm is essential [6]. The main goal of the study is to suggest an effective method for changing the shift rows and mix column stages of AES coupled with ECC. Compared to different methodologies, ECC can offer a similar degree of safety with a more modest key size. The creation of a system that uses the cloud to provide data security at a lower computational cost and a faster encryption/decryption procedure is necessary [31-34].

To utilize the characteristics of the two strategies, we blend them in our recommended model. The main findings of the study are as follows. We propose a hybrid paradigm that utilizes ECC for AES key creation and combines AES and ECC. The four sub operations of ECC-EAESKDS are Sub Bytes, Symmetrical Transposition, Bitwise Reverse Transposition and Add Round Key Operation. In our proposed method, ECC-EAESKDS 1st, 2nd and 3rd stages of AES are substituted by Key Dependent S Box generation using irreducible polynomials and affine transformation, Symmetrical Transposition and Bitwise Reverse Transposition, respectively. A 256-bit key and a 128-bit plaintext block (data) are used as inputs by the algorithm.

Similar to symmetric/asymmetric encryption, data encryption and decryption require either a private key or a public key. Due to the tremendous key size expected by this strategy, it requires a great deal of computing power. The ECC-EAESKDS hybrid strategy that has been proposed deals with the issue of the key's size while using fewer computer resources for memory optimization, which has the advantage of hastening the increase in system security.

The sections that follow make up the rest of this paper. Studies that are related are provided in Section 2. In Section 3, the research technique is explained in detail. In Section 4, the results and observations of the experiment are discussed. In Section 5, the work's decision is introduced.

II. Literature Survey

Because all users share resources in sync, cloud storage is becoming more and more popular. Because cloud storage is always accessible, data owners choose it over other providers. Data integrity and data preservation should be examined for this reason to enhance system security. To improve system security, AES and ECC are suggested [7]. Without a trusted center, the system is distributed and managed using Shamir secret sharing. The combination strategy that has been offered does increase system security, but at a significant computational cost and time expense.

Along with the suggested technique, which makes use of comparable algorithms for secure cloud services, AES, DES, and Blowfish are also used [8]. These algorithms offer data storage efficiency and integrity to prevent conflict between large groups of users and individually secure the data of each user. In addition, the service provider manages and expedites data access. Cloud computing data services also measure the size of data blocks and the avalanche effect of plain text. The security advantages of ECC with RSA can be compared in a study [9] by Madhavi et al. using data larger than 264 bits, as 256-bit data exceed the NIST restrictions. Due to providing higher safe services over smaller data volumes and reduced storage requirements for data accessibility, the ECC approach outperforms the RSA method in this performance comparison.

Hybrid approaches for RSS and ECC are used in the study [10]. After the data has been compressed, the elliptical curve authorities receive some pieces that need to be signed so they can digest and sign them. The same method is used to carry out the encryption process. The supremacy of RSS and ECC analysis provides the basis for the development of hybrid algorithms. Providing secure and private data security is a major challenge for cloud computing services [11, 12]. Due to privacy concerns, we are unable to keep raw data without encryption because the CSP is an untrusted third party. The proposed study examines a hybrid cryptosystem-based solution for the transport of reliable and reliable information in the cloud. By simultaneously implementing AES and ECC to enhance the framework's categorization and credibility, we may use symmetric and divergent encryption to enhance the cloud data security. As a result, the projected model manages an efficient, powerful, and safe encryption approach based on AES and ECC.

The capabilities of cryptography for providing security in distributed storage were introduced in papers [13, 14]. This was achieved by investigating common cryptography methods such as AES, ECC, and RSA. The question of identifying an efficient and secure encryption approach was, in any case, settled by this investigation with regard to the variances in the display of these operations. Although certain encryption techniques can guarantee security, they take a long time to encode and decrypt data. However, while various solutions may provide effective encryption, they nonetheless suffer from the negative impacts of the requirement for security.

Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA) and PRESENT algorithms [15] are only a few encryption techniques that rely on s-boxes to strengthen the diffusion process. In the past ten years, the Advanced Encryption Standard has been the subject of increasing research. Numerous applications, particularly those that involve media, such as photos [16] and movies [17], employ AES as a key encryption technique. According to the biometric system, E. Siva Ganesh et al. [18] suggested a method for producing dynamic s-boxes. They demonstrated how the greater nonlinearity of the recommended s-box results in a stronger AES system [19]. Their approach is suitable for encryption applications due to its speed. However, both the encryption and decryption processes must take into account the biometric technique. Kamsiah Mohamed et al. [20] and Hui Shi et al. [21] investigated the s-box's nonlinearity effect of the s-box in the AES encryption procedure. Since the s-box is the sole stage in the process that adds non-linearity impact, the effect must be strong enough to prevent hacking by attackers. They both came to the conclusion that, in order to evaluate the strength of the s-boxes, the avalanche effect must be looked at more thoroughly than any other factor.

Data encryption is altered non-linearly by the substitution process in the AES encryption method. By including the confusion feature into the algorithm, it satisfies one of the key conditions for data security as outlined by Shannon

(Shannon, 1998). Repetition of transposition with substitution, according to Shannon, produces secure ciphers. Early cryptographic systems employed substitution permutation networks (SPNs) to combine substitution and permutation circuits. The encryption process must, however, be strengthened by at least one non-linear process in contemporary cryptographic systems [22]. The s-box, which demands the same length for both the message and the key, is the most crucial stage in the AES algorithm, as was previously explained. However, the s-box replacement step causes the most latency in the AES algorithm [23]. In papers [24, 25], a strategy to improve cloud computing information security and a two-level cryptographic method were presented. By enhancing information security against intrusions by utilizing both symmetric and uneven encryption calculation (AES and ECC), prevent them from accessing the actual information, empower privacy, respectability of the information, and time required to perform cryptographic tasks, the model increases client confidence in cloud computing and accelerates the use of smaller ECC keys in cryptographic interactions.

III. Proposed Frame Work

In this section, we provide a comprehensive design description of the proposed scheme. We emphasize the importance of combining AES and ECC and the calculation used in this plan.

A. Definition of ECC and AES

By employing the subsequent asymmetric key encryption, the data are protected from unauthorized access by the well-known cryptographic technique known as ECC. The ECC's security is safeguarded by using key pairs that are both public and private. As prime and binary fields, ECC makes use of two-dimensional fields. The use of enhanced operations and the creation of a connection between binary and primary fields that prevents unauthorized access, make hacking difficult with this cryptographic strategy. The small key size of the ECC is a crucial feature. When the maximum number of points is used to find the right field during the cryptographic implementation, this can provide enhanced data security.

Before generating a massive number that can range from 0 to Z based on the input, the field's start action selects the first number. Because ECC is used exclusively to generate the key, the processes are simplified. Because of its tiny key size, ECC has a substantially larger enhancement than other cryptographic techniques. In this study, the enhancement of memory and space is optimized using ECC techniques [21]. One cipher text format that uses block ciphers is AES. This protects your data by encrypting and decrypting it with just one key. This includes various performance activities constrained by cloud storage, such as cloud storage retrieval and statistical analysis. Security policies for cloud storage are most often enforced by the strategic algorithm in cloud computing. This article uses the AES encryption method because it is simple to set up and compatible with data that can be recovered from cloud storage [26, 27].

B. ECC and Improved – A Suggested Combination Hybrod Method

The most advanced and effective encryption technique for cloud storage was developed using ECC and AES. AES uses a larger key size, so it is slower than the hybrid (ECC-EAESKDS) method, since the hybrid model permits a smaller key size and a quicker security mechanism to safeguard the data. The hybrid model ECC-EAESKDS uses a decrease in key size for encryption and boosts performance because the tiny key size of ECC is its main point of differentiation [28]. The ECC has set standards for encryption and decryption keys to create a safe key system and minimize key size.

ECC and AES work well together to encrypt data and block unauthorized access. Data encryption and decryption produce cipher text after selecting the key size. AES uses the key that ECC generates. The proposed cloud storage approach can provide a safe system due to the combined impacts of ECC and AES. In this way, the size of secure data storage can be decreased.

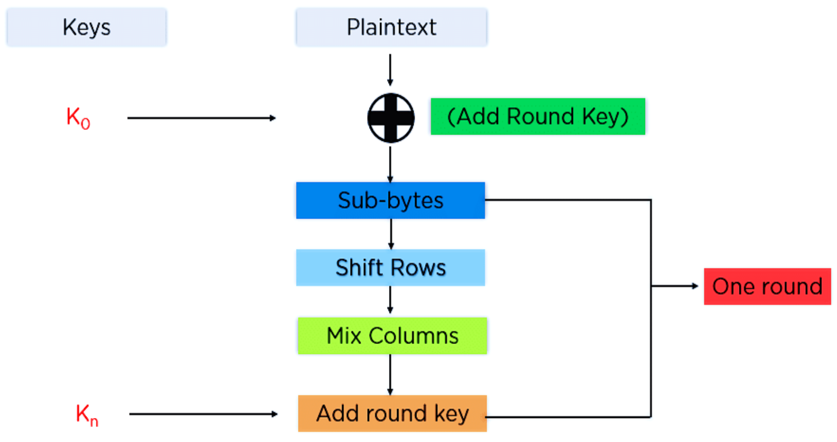

Fig. 1: Workflow for AES Encryption Algorithm

The length of the key affects the number of rounds in AES. AES employs 10 rounds for 128-bit keys. 12 rounds for a key with 192 bits. For 256-bit keys, there are 14 rounds, as seen in Fig. 1. Each encryption round consists of four phases. Byte Substitution (Sub Bytes), Row Shift, Column Shuffle, and Round-Key Addition. The decryption process works exactly the opposite way. Security is not compromised as long as the system is designed effectively and uses effective key management methods.

i) Add round keys;

The Add Round Key operation functions on the AES Round key similarly to how AES encoding works. The input to the round is exclusively-ored with the round key throughout this procedure. In this study, the Add Round Key operation of ECC-EAESKDS functions on the AES Round key similarly to how AES encoding works. Throughout this process, the round key will only be used to input data into the round.

ii) Sub Bytes

Changing a state array byte by consulting a substitution box (also known as S-box) lookup table is referred to as Sub Bytes. The S-box is a 16*16 lookup table with 256 unique values. Every possible value for an 8-bit sequence, or 0 to 255 in decimal, is represented in the S-box table. A byte from the state array is input into this Sub Bytes step, which then replaces it with a value that matches. The encrypted information can be predicted without the encryption key if the avalanche effect isn't satisfied to a certain degree, which causes insignificant randomization. Regarding symmetric key cryptographic algorithms, it is desired to acquire the strict avalanche criterion. In a block-cipher context, the SAC is said to be maintained by algorithms if, when one bit in the key or plaintext is complemented, the cipher text is significantly altered about one half of the cipher text. The algorithms' confusion and diffusion properties are wholly dependent on this SAC. The Sub Bytes, Shift Rows, and Mix Columns steps all contribute to a desired level of confusion and diffusion in the context of the AES symmetric key.

iii) Symmetrical transformation

The proposed ECC-EAESKDS symmetrical transposition part replaces the shift row of AES in this section, and all rows and columns of the cipher's internal 128-bit state are combined. Wherever there is a byte in every cell, that may be a 4x4 matrix. The positions of non-major diagonal sections that are symmetric with regard to the main diagonal components are switched using inner-state bytes. The rows and columns of the matrix are filled with these bytes. Then, in reference to the non-diagonal array of elements, the primary symmetrical diagonal components are reversed.

iv) Bitwise reverse transformation

The bitwise reverse transposition stage offers diffusion by mixing the input round, just as the symmetrical transposition phase of the proposed ECC-EAESKDS. Bitwise reverse transposition, as opposed to the mix columns section of the original AES, uses optimal operations to speed up the encoding and secret writing algorithms as shown in the below algorithm.

Algorithm 1: The Proposed Framework Algorithm

Using Elliptic Curve Cryptography [29] to Create Public Keys.

Step I. Consider a prime number n.

Step II. To produce the public key, we can use any of the numbers n(a), where n(a) > n.

Step III. In cases where G > n, determine the curve's point as G.

Step IV. The public key is calculated using the equation P = n(a) * G.

Step V. The calculated public key P is returned.

Enhanced Advanced Encryption Standard Encryption and Decryption with Key Dependent S Box

Step 1: First, open the input file.

Step 2: Next, add the public key produced by the ECC.

Step 3: Using the public key produced by ECC, ECC-EAESKDS encryption is carried out on the input file.

Step 4: The ECC-EAESKDS round key operation uses the AES round key in a manner similar to that of AES encoding.

Step 5: The S-Box, also known as a substitution table, is the non-linear component that makes up the basic core of sophisticated encryption standards. The S-Box should be built in such a way that it can withstand algebraic assaults like differential and linear cryptanalysis. AES has a non-linear layer called the S-Box that causes misunderstanding. S-Box, which is utilized for decryption, must be invertible. The S-Box is nothing more than 256 8-bit values permuted. All plain text input bytes in S-Box are mapped to, or replaced with, new bytes that match each other. The S box in AES is generated using the irreducible polynomial $x^8+x^4+x^3+x+1$ and $GF(2^8)$ (Galois Field). The values of the rows and columns in the AES S-Box matrix, which has 16 * 16 = 256 elements, range from 0 to 15 (0 to f in hexadecimal notation). Each byte of the S-Box in $GF(2^8)$ is mapped to its multiplicative inverse. The methods listed below are used to generate the key-dependent S box for every AES cycle. The algorithm is supposed to select an irreducible polynomial from among the 30 available irreducible polynomials, an affine constant from among all affine values between 0 and 255, and a key EX-OR value that signifies the EX-OR of all key bytes whenever a single bit of a key is complemented.

Procedure for 1st round

1. Determine the secret key value k, which is dependent on the entire key values of wr(i).
2. k = ⨁_(i=1)^j AESSBox(w1(j))
3. Depending on the key value k, choose the irreducible polynomial m from the IrrPoly array:
4. I = k mod 30 and m = IrrPoly(i). Using the AES S-Box, choose the affine constant c that is dynamically based on the key value k. C will have a value between 0 and 255.
5. c = AESS-Box(k)
6. for all j=0, 1, . . . , 255 do :

a. Using the irreducible dynamic polynomial m, get the p inverse of each element of j:
b. Using the AES S-Box, Choose a constant whose value is dynamically dependent on the key value k. C will have a value between 0 and 255:
c. p = FindInv(j,m)
d. Using an affine constant c that is dynamically chosen, apply the affine transformation to all values of p:
e. t = AffiTrans(p, c)
f. Make the creation of S-boxes reliant on the key value k now.
g. SBox(j) = t ⨁ k

7. end for
8. DySBox=S-Box
9. Generate key-dependent dynamic inverse InvDySBox:
10. for all j=0, 1, . . . , 255 do :

```
InveDySBox(DySBox(j)) = j
11.     end for
```

Procedure for 2nd round to r-1 rounds

Calculate the secret round key value of the round kr using the dynamic S-Box formed by the previous round, which depends on the full key values of the key wr(i) for round r.

```
1.      for all r=2, . . . , r - 1 do
a.      r=⨁_(i=1)^jDSBoxr(wr(j))
b.      Depending on the key value kr for round r, choose the irreducible polynomial mr from
        the array IrrPoly:
c.      I = k mod 30 and m = IrrPoly(i). choose a constant whose value is dynamically depending
        on the key value kr. cr will have a value between 0 and 255:
d.      c = DySBoxr(k)
          for all j=0, 1, . . . , 255 do :
i.Using the dynamic irreducible polynomial mr, get the p inverse of each element of j:
i.Using the AES S-Box, choose a constant whose value is dynamically dependent on the key value
  k. C will have a value between 0 and 255:
i.p = FindInv(j,mr)
v.Using an affine constant cr that is dynamically chosen, apply the affine transformation to all values
  of p:
v.t = AffiTrans(p, cr)
i.Generate DSBoxr for round r, which is dependent on the round key value kr, to generate the S box
  for round r.
i.SBox(j) = t ⊕ kr
e.      end for
2.      end for
3.      Fill up the DySBox with new S-Box values for round r:
4.      DySBoxr = SBoxr
5.      Create a dynamic inverse that is key-dependent. InvDySBox
6.      for all j=0, 1, . . . , 255 do :
InveDySBox(DySBox(j))= j
7.      end for
```

Step 6: During the shift rows phase, the positions of the nonmajor diagonal elements that are symmetric with respect to the main diagonal should be swapped, and the members of the major diagonal should then be reverse

Step 7: During the mix column phase, we transfer the elements from the input's first row to the output's first column, then switching a21 for a31 and reversing the bit order in the result. Taking elements from the second row of the input and moving them to the third column in the output, then switching a23 to a33 and flipping the bit order in the output. Taking the components from the third row of the input and moving them to the second column in the output, then swapping a22 and a32, and flipping the bit order in the output. Taking the components from the fourth row of the input and moving them to the fourth column in the output, then swapping a24 and a34 and flipping the bit order in the output.

Step 8: The file is then uploaded to the server and encrypted with ECC-EAESKDS. After being uploaded, the original file's encryption is broken using the translation utilizing the ECC's public key.

To decipher the recommended ECC-EAESKDS encrypted cipher text, each stage of the encryption operation must be reverted in the opposite order it was utilized. It is clear that ECC-EAESKDS successfully safeguards data when stored in the cloud. Encrypted data ensure the safe transfer of user data to the server and the subsequent storage mechanism. Innovation can also be evaluated using costs and computation time.

IV. Results and Discussions

Based on the following security assessment criteria, our suggested technique, ECC-EAESKDS, is created and contrasted to the original AES algorithm and the ECC-EAESKDS algorithm: Hamming distance, avalanche effect, mathematical soundness, encryption and decryption times, and avalanche and diffusion effects. The execution makes use of an 8 GB of memory and a 2.7 GHz Intel Center i5 processor. These techniques were examined using the Amazon AWS SDK and the Amazon Simple Storage Service. A 256-bit key and a 128-bit plaintext block (data) are used as inputs by the algorithm.

The suggested ECC-EAESKDS algorithm includes properties that increase security by making the system more complicated and resistant to attacks. Furthermore, it is clear that the suggested h technique encrypts and decrypts data much faster than existing methods. Our computational cost decreases together with the time required for encryption, which is incredibly efficient. As can be shown in Figs. 2 and 3. Our suggested method is therefore more effective than others.

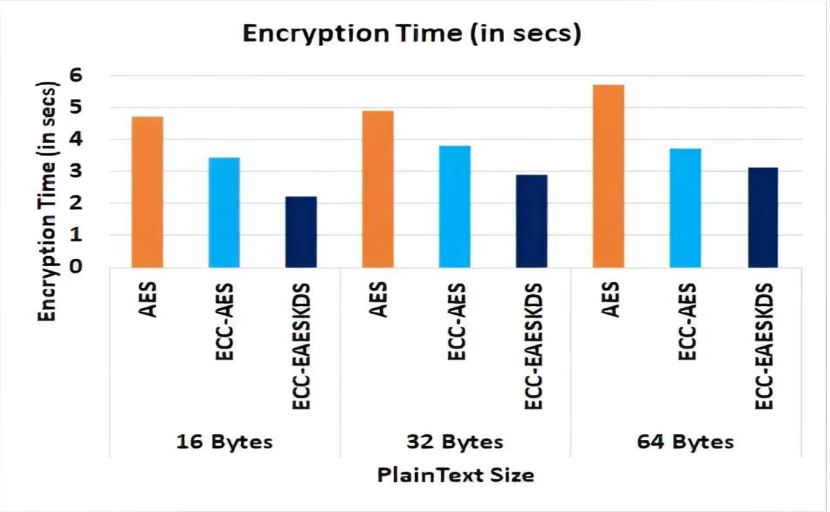

Fig. 2: Comparison of encryption time of different algorithms

How much an algorithm has changed can be determined using the Avalanche Effect (AE). Simply put, it means that the output of the text can be significantly affected by a minor change in the input. By adding the modified bits to the cipher bits and dividing the result by the entire amount of cipher bits, we get the AE. The equation we used to calculate the Avalanche Effect

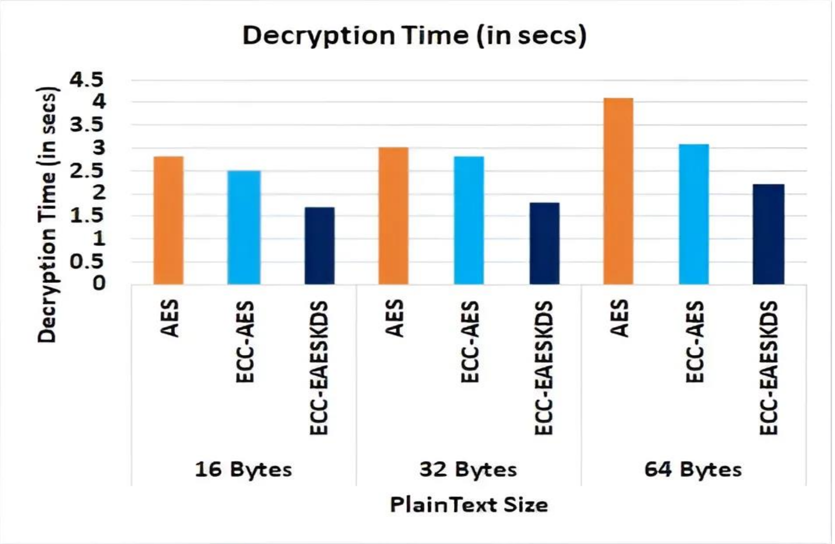

Fig. 3: Comparison of decryption time of different algorithms

AE = change in bit count / total bit count

We contrasted the Avalanche effect of our proposed strategy with that of the AES algorithm and the ECC-AES algorithm, as illustrated in Fig. 4. If an algorithm's Avalanche impact is greater than 50%, it is regarded to have greater security strength than others. Thanks to our suggested method's ability to achieve the maximum Avalanche effect, the system is safer than others.

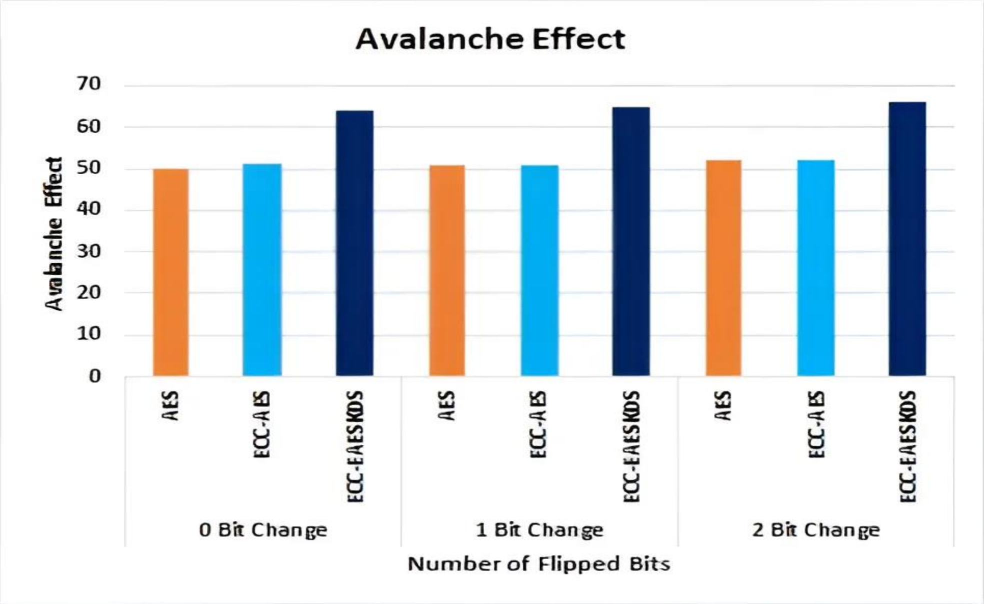

Fig. 4: Avalanche Effect Comparison of Different Algorithms

Block ciphers and cryptographic hash functions both heavily depend on the avalanche effect. A cryptosystem has the avalanche property if the output changes dramatically (about half of the output) after flipping or changing just a single bit in plaintext or in the secret key. It is employed to determine how changes to the output bits will affect the input bits. In other words, it counts the number of output bits that flip when an input bit flips. A slight modification to the input data or encryption key causes a drastic change in the encrypted text in the output for strong cryptographic algorithms. Failure to meet the avalanche effect's requirements to a certain degree results in little randomization and makes it possible to anticipate encrypted data without knowing the encryption key. When it comes to symmetric key cryptography algorithms. The desired property is the strict Avalanche criterion. In the context of a block cipher, the SAC is said to be preserved by algorithms if the addition of just one bit, either in the plaintext or the key, results in a significant change in the cipher text roughly one-half of it. This SAC is entirely dependent on the algorithms' properties of confusion and diffusion. Sub-bytes step, shift row step, and mix column step provide the desired level of uncertainty and diffusion in an AES symmetric key environment.

Diffusion, a feature of the avalanche effect in cryptography, displays the strength of a method from a cryptographic perspective. Any change to the associate degree input, no matter how modest, has a significant impact on the final result (plaintext or secret key). The avalanche effect is another name for this. We used the Hamming distance to calculate the avalanche effect. The Hammer distance is a tool used in information theory to measure dissimilarity. Because it is simple to implement programmatically, as the sum of bit-by-bit xor (exclusive or) considering ASCII values, we get the Hamming distance. It is recommended to have a high diffusion rate or high avalanche outcome. The avalanche findings show the performance of a cryptographic algorithm. The avalanche effect increases with the number of variations in the cipher that result from a single bit change in the key or plain text. We can observe that the harder it is to simply break the algorithm, the larger the avalanche effect, as with our suggested solution. We can observe that our solution has improved security due to the avalanche effect.

In this section, we compare the diffusion characteristics of the shift rows of the AES with those of the symmetrical transposition method. The diffusion attribute is calculated using the Hamming Distance (HD), which is the distance between two strings of the same length. Both the measured Hamming distance between input and output (current output) and the change in cipher value were tested during the execution of the two previous procedures.

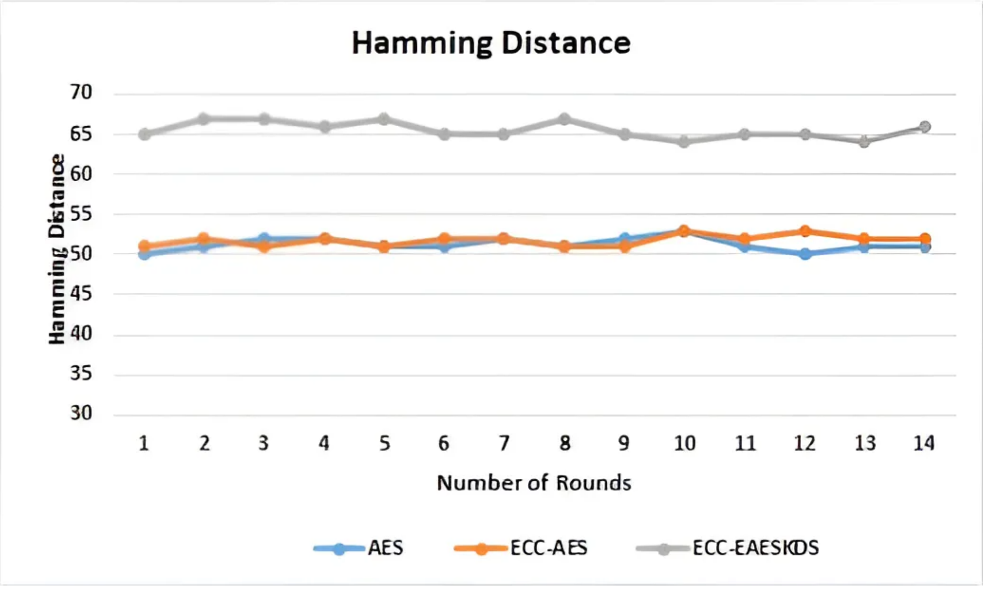

Fig. 5: Comparison of Hamming Distance with Different Algorithms

According to the results shown in Fig. 5, the proposed method's hamming distance values range from 58 bits to 74 bits, whereas those of the ECC-AES and AES algorithms were, respectively, 57 bits to 71 bits and 58 bits to 78 bits. The less diffusion property is produced when a shift row operation exchanges positions with a sub operation.

The single non-linear element in any cryptographic method is the S-box. An S-box's capacity to withstand both linear and differential cryptanalysis is what makes it nonlinear. The security of an S-box will increase as the value of nonlinearity increases. With the AES S-box, the greatest nonlinearity value attained is 112. Since it is currently the norm for all other S-boxes, those that have this nonlinearity value are often regarded as secure. The S boxes suggested in the study achieved the maximum non-linearity score, or 112. The SAC predicts that a single bit change in the input will alter half of the output bits and its value should be close to 0.5. The SAC value for the study's suggested S boxes was 0.501. According to the bit-independence criteria, the output will change separately for each given change in the input bits. All of the functions must be nonlinear in order for the S-box to satisfy the BIC and the SAC. In our example, the suggested S-boxes with a value of 112 also satisfy this condition. As a result, the suggested enhanced hybrid AES ECC with key-dependent S boxes exhibits improved performance.

V. CONCLUSIONS

The infinite infrastructure for mobile applications offered by cloud computing allows for great scalability, low maintenance online data execution. It is clear that with the implementation of cryptographic techniques for safe computation, the ongoing vulnerability in the cloud may still be managed. Data encryption protects data from unauthorized users, ensuring security and data confidentiality. Data privacy is guaranteed using a hybrid encryption scheme. The suggested model made use of the fast-symmetric scheme and less computationally demanding resilient cryptosystem methods of the AES algorithm with its key encryption using ECC. As previously noted, we put forth the two proposed methods of bitwise reverse transposition and symmetrical transposition. These suggested methods were created for the ECC-EAESKDS algorithm. The present sub-bytes are replaced by key dependent dynamic S box using irreducible polynomial and affine transformation for achieving better diffusion and confusion. The AES shift row step was replaced by symmetrical transposition based on the suggested techniques. Additionally, AES's bitwise reversed transposition was used in place of the AES mix columns stage. To balance security and encryption time, this was done to speed up the encryption method.

REFERENCES

[1] Alexandra Durcikova Murray E. Jennex. (2017). Introduction to Confidentiality, Integrity, and Availability of Knowledge and Data Minitrack. Proceedings of the 50th Hawaii International Conference on System Sciences, 2017, Page - 4287.

[2] Altatar, M. A. (2017). Modified Advanced Encryption Standard Algorithm for Reliable Real-Time Communications, International Journal of Computing and Digital Systems, Pages 303-309.

[3] Awad, A. I. (2018). Introduction to information security foundations and applications. Research Gate, Retrieved from https://www.researchgate.net/Publicati on/325170901.

[4] Ayushi Arya et al. Review Paper on Effective AES Implementation, International Journal Of Engineering And Computer Science ISSN: 2319-7242, Volume 4 Issue 12 Dec 2015,Page No. 15403-15405.

[5] Avi Kak, AES: The Advanced Encryption Standard, Avinash Kak, Purdue University, January 31, 2019, page 20-11.

[6] Rizky Riyaldhia, et al, (2017., October 13-14). Improvement of advanced encryption standard algorithm with shift row. Elsevier B. V., ScienceDirect, Procedia Computer Science 116 (2017), pages 401–407.

[7] Shukla, D.K.; Dwivedi, V.K.; Trivedi, M.C. Encryption algorithm in cloud computing. Materials Today Proc. Volume 37, Part 2, 2021, Pages 1869-1875.

[8] Yahia, H.S.; Zeebaree, S.R.M.; Sadeeq, M.A.M.; Salim, N.O.M.; Kak, S.F.; Al-Zebari, A.; Salih, A.A.; Hussein, H.A. Comprehensive survey for cloud computing based nature-inspired algorithms optimization scheduling. Asian Journal of Research in Computer Science, May 2021, Page 1-16.

[9] Qazi, R.; Khan, I.A. Data security in cloud computing using elliptic curve cryptography. International Joutnal of Computing and Communications Networks, ISSN: 2664-9519 (Online);Vol. 1, Issue1, August 2019, 46–52.

[10] Manaa, M.E. Data encryption scheme for large data scale in cloud computing. Journal of Telecommunication, Electronic and Computer Engineering • September 2017. 9, 1–5.

[11] Arockia, P.; Dharani, N.; Aiswarya, R.; Shailesh, P. Cloud data security using elliptic curve cryptography. International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 04 Issue: 09 | Sep -2017.

[12] Li, Y.; Gai, K.; Qiu, L.; Qiu, M.; Zhao, H. Intelligent cryptography approach for secure distributed big data storage in cloud computing. Information Sciences, Vol. 387, May 2017, Pages 103-115

[13] Saeed, Z.R.; Ayop, Z.; Azma, N.; Rizuan Baharon, M. Improved cloud storage security of using three layers cryptography algorithms. Journal of Computer Science IJCSIS. 2018, 16, 34–39.

[14] Al-Dhuraibi, Y.; Paraiso, F.; Djarallah, N.; Merle, P. Elasticity in cloud computing: State of the Art and research challenges. IEEE Transactions on Services Computing. 2017, 11, 430–447.

[15] Jithendra, K. B., & Shahana, T. K. (2019). A New Efficient Sbox for Strengthening PRESENT Like Block Ciphers Against Linear Cryptanalysis. 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies, ICICICT 2019, 507–511. https://doi.org/10.1109/ICICICT46008.2019.8993397.

[16] Alsaffar, D. M., Sultan Almutiri, A., Alqahtani, B., Alamri, R. M., Fahhad Alqahtani, H., Alqahtani, N. N., Mohammed alshammari, G., & Ali, A. A. (2020). Image Encryption Based on AES and RSA Algorithms. 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS), 1–5. https://doi.org/10.1109/ICCAIS48893.2020.9096809.

[17] Adiguna, T., & Hendrawan. (2017). Secure H.264 Video Coding using AES/CFB/PKCS5 padding encryption on various video frames (I, P, B). Proceeding of 2016 10th International Conference on Telecommunication Systems Services and Applications, TSSA 2016: Special Issue in Radar Technology, 5–9. https://doi.org/10.1109/TSSA.2016.7871104.

[18] Ganesh, E. S., Velayutham, R., & Manimegalai, D. (2012). A secure software implementation of nonlinear AES S-box with the enhancement of biometrics. 2012 International Conference on Computing, Electronics and Electrical Technologies, ICCEET 2012, 927–932. https://doi.org/10.1109/ICCEET.2012.6203796.

[19] Al-Dweik, A. Y., Hussain, I., Saleh, M. S., & Mustafa, M. T. (2019). A Novel Method to Generate Key-Dependent S-Boxes with Identical Algebraic Properties. 1–20. http://arxiv.org/abs/1908.09168.

[20] Mohamed, K., Mohammed Pauzi, M. N., Hj Mohd Ali, F. H., Ariffin, S., & Nik Zulkipli, N. H. (2014). Study of S-box properties in block cipher. I4CT 2014 - 1st International Conference on Computer, Communications, and Control Technology, Proceedings, I4ct, 362–366. https://doi.org/10.1109/I4CT.2014.6914206

[21] Shi, H., Deng, Y., & Yu, G. (2011). Analysis of the avalanche effect of the AES S box. 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce, AIMSEC 2011 - Proceedings, 5425–5428. https://doi.org/10.1109/AIMSEC.2011.6009935.

[22] Hosseinkhani, R. (2012). Using Cipher Key to Generate Dynamic S-Box in AES Cipher System. International Journal of Computer Science and Security (IJCSS), Volume (6) : Issue (1) : 2012, 6, 19–28.

[23] Hodowu, D.K.M.; Korda, D.R.; Ansong, E.D. An enhancement of data security in cloud computing with an implementation of a two-level cryptographic technique, using AES and ECC algorithm. International Journal of Engineering Research & Technology (IJERT), Vol. 9 Issue 09, September-2020, 639–650.

[24] Zhu, Y.; Fu, A.; Yu, S.; Yu, Y.; Li, S.; Chen, Z. New algorithm for secure outsourcing of modular exponentiation with optimal checkability based on single untrusted server, Proceedings of the 2018 IEEE International Conference on Communications (ICC), 20–24 May 2018; pp. 1–6.

[25] Bhardwaj, K.; Chaudhary, S. Implementation of elliptic curve cryptography in ‘C’. International Journal on Emerging Technologies 3(2): 38-51 (2012).

[26] Ogiela, U. Cognitive cryptography for data security in cloud computing. Concurrency Computation Pactice and Experience, 2019, 32, e5557.

[27] Sood, S.K. A combined approach to ensure data security in cloud computing. Journal of Network and Computer Applications, Volume 35, Issue 6, November 2012, Pages 1831-1838.

[28] Mendonca, S.N. Data security in cloud using AES. International Journal of Engineering Research & Technology (IJERT), Vol. 7 Issue 01, January-2018.

[29] R. S, S. Bhargavi, B. G. M, S. K. N, P. Chavan, and S. S, “Image Encryption and Decryption using Symmetric Key Sequence of Elliptic Curve (EC) Over Prime Field,” 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT). IEEE, Oct. 20, 2023. doi: 10.1109/easct59475.2023.10392967.

[30] Y. M. Dalal, S. N. Raj, S. S, S. G, Y. T, and A. Biradar, “Comparative Approach to Secure Data Over Cloud Computing Environment,” 2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS). IEEE, Nov. 02, 2023. doi: 10.1109/csitss60515.2023.10334187.

[31] T. Yerriswamy and M. Gururaj, “An Efficient Hybrid Protocol Framework for DDoS Attack Detection and Mitigation Using Evolutionary Technique,” Journal of Telecommunications and Information Technology, vol. 4, no. 2022. National Institute of Telecommunications, pp. 77–83, Dec. 29, 2022. doi: 10.26636/jtit.2022.165122.

[32] Yerriswamy T and Gururaj Murtugudde. Signature-based Traffic Classification for DDoS Attack Detection and Analysis of Mitigation for DDoS Attacks using Programmable Commodity Switches [J]. Int J Performability Eng, 2022, 18(7): 529-536.

[33] Y. T and G. Murtugudde, “An efficient algorithm for anomaly intrusion detection in a network,” Global Transitions Proceedings, vol. 2, no. 2. Elsevier BV, pp. 255–260, Nov. 2021. doi: 10.1016/j.gltp.2021.08.066.

[34] Y. T and G. Murtugudde, “Study of Evolutionary Techniques in the field of Network Security,” 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE). IEEE, Oct. 09, 2020. doi: 10.1109/icstcee49637.2020.9277082.