Performance Analysis of Hybrid Cryptography System for High Security and Cloud-Based Storage

R.BHAGYALAKSHMI, ROOPASHREE D., SHRUTHI.K. N
Department of Electronics and Communication, Government Engineering College, Hassan, INDIA

Abstract: - Now a day, the security of the data is playing a major part in communication systems due to further bushwhackers between channels media. The security position depends on cache crucial and as per literature, advanced the bit size of the key, advanced the security and also larger data size comes major challenge task for the further process. Thus, the generation of crucial with the further size is a major grueling task and at present, the Advanced Encryption Standard (AES) is a better cryptography system where the encryption and decryption can perform with a fixed key. The literature says the holomorphic function is well-suitable for data size reduction. To address this issue, new Holomorphic grounded encryption and decryption and AES are combined to increase the security position. The alternate novelty is that variable crucial generation using Elliptic Wind Cryptography (ECC) due to its enlarged proportion of consideration in assiduity and experimenters. The ECC uses point addition and point doubling to induce 256 values and addition operations can be avoided. After the generation of the matrix, each matrix value is translated and decrypted using a Holomorphic algorithm. The proposed work has been designed using MATLAB 2017a, dissembled, and validated with different datasets in real decors. Cloud computing is expected to be considered one of the primary computing platforms in the field of storage and security as it possesses many advantages such as profitability, efficiency as well as lower implementation overheads. Contemporary cloud computing security algorithms are enhanced extensions of cryptography. Data privacy, as well as data protection, are the major areas of concern in Cloud computing. The cryptographic with holomorphic based data encryption and interchange of information is exchanged and then accumulated in the cloud through holomorphic encryption which uses point addition and doubling operation to ensure data confidentiality of owners as well as users. Proposed work novel hybrid algorithm based on the context of encryption and decryption and thus integrates cryptography hybrid techniques include modified 126-bit AES and ElGamal based ECC through splitting algorithm. The advantage of splitting the larger data in size into binary form and then processing for encryption and decryption leads to optimization of latency, increase throughput, and security. The proposed hybrid approach has better security towards information sharing as well as cloud storage intrusions. Based on obtained results in MATLAB 2017a software tool, the obtained results show that 43% improvement in throughput and 12% reduction in latency, and a 21% improvement in security level.

Key-Words: - AES, Cloud computing, ECC, Holomorphic, Elgamal algorithm and point addition and point doubling.

Received: May 29, 2021. Revised: June 2, 2022. Accepted: June 26, 2022. Published: July 20, 2022.

1 Introduction

Many security challenges persist in the existing techniques for storing data in the cloud. Data breaches, cloud authorization, multi-tenancy, internal threats, and improper handling are some of the challenges faced during cloud computation and it is difficult to monitor security measures that meet the security requirements of all users. This issue is because different users seem to have different security issues depending on the application of cloud-based services. In the case of the owner as well as the user, the cloud service provider (CSP) offers an excellent security layer. The user must ensure that there is no information inadequacy or unauthorized disclosure of data for other users who are using the same cloud. Thus, CSPs must be trained well to deal with cyberattacks. Information security is not guaranteed by all cloud suppliers. Various approaches in cloud data storage have been used to eliminate the problems concerned with security [1]. Generally, data security concepts mainly focus on border security as well as on networks along with the application of tools like intrusion-detection modules as well as firewalls. When compared to APTs, special users, as well as other malicious types of security threats, this technique does not provide adequate security. To guarantee the protection of the key, the encryption process must be integrated with a reliable key management platform [2]. It is extremely important

to evaluate the complete encryption as well as key management procedure. Encryption integrates with other primary data security improvements to provide an inclusive-hybrid technique to handle confidential information in and out of the cloud and thereby imparting enhanced data safety [3]. To ensure the information in the cloud and to provide the essential level of security, any strategy that is based on data must include encryption, key administration, minimized access limits, as well as security procedures. [4] By combining these key aspects in a hybrid strategy, the companies can improve their security more effectively as well as extensively rather than depending just on conventional organization security mechanisms[5].
The following are the key objectives of the suggested work:

- Improving data security in the cloud by introducing a unique methodology.
- Addressing the Cloud Computing security problems.
- To devise a more effective approach for ensuring secure data exchange in the cloud.

The suggested work is primarily organized into six sections. The first section covers an overview of the research work as well as a brief description of the main concepts. The following section discusses the literature survey of the previous work that is used as experimental information for the future as well as the summary of the novel design methodology. Section three explains the suggested technique as well as the algorithm used. The simulation concepts, as well as the structure concept, were discussed in the fourth section. The outcomes of the simulation parameters that were evaluated in section three are reported in the fifth section. The final section of the article presents the conclusion of the suggested work.

2 Literature Review

A method of encrypting data to keep it hidden from unauthorized users is termed Cryptography [6]. Transmitted data is clouded as well as sent in an obscure as well as an inaccessible form of ciphertext form to an unauthorized individual. A key is used to convert the encrypted message into plain text. This key is kept confidential and only authorized users can access the key [7]. Encryption is probably the best strategy for avoiding Malicious activities because the hacker will not be able to decrypt the data even if it is denied. Certain highly secure strategies probably cannot be cracked with infinite processing power such as the one-time pad. Consequently, these strategies are more difficult to execute than the theoretically hackable but practically secure systems.

With the help of stenography, image encryption-decoding is carried out by employing the Shamir approaches [8]. The actual image is scrambled in the first stage using Shamir methods and this strategy is used to encrypt information. Data is encrypted using border scanning approaches in such a way that hackers cannot hack the secret information. The technique presented in [9] employs two secret keys to encrypt as well as decrypt the data. One secret key is used for the background image and the other is used for the information. Initially, the image is transmitted and then encrypted. At last, the encrypted image is decrypted using a secret key. AES employs a variety of key sizes namely 256-bit, 192-bit, as well as 128-bit [10]. With the application of basic digit shrewd XOR operation between state and round key, the sub bytes transformation, Shift row transformation, Addition of round-keys, as well as Hybrid-columns are executed.

Since RSA presented in [10] is more effective and possesses a faster AES technique, this strategy is preferable for evaluating asymmetric as well as symmetric-cryptograph. Asymmetric cryptography can be employed to authenticate users as well as choose a symmetric-encryption key. Large data blocks are securely authorized using the RSA technique as well as encoded using the efficient AES computation rather than the slow RSA computation. Elliptical curve cryptography that is based on the public-key cryptography method is presented in [11] which is built on the elliptic curve hypothesis. Authentication of ECC is more effective for remote trades such as mobile phones, and smart cards as well as personal data such as money transactions or confidential clinical reports, and classified data, where providing confidential data is the major concern in this work.

3 Problem Formulation

Based on the literature check bandied for different algorithms about encryption and decryption process for adding security position, it plants that utmost of the cryptography systems uses lower crucial sizes with lower of number computational operations. Utmost security systems use kindly encryption of holomorphic structure recycling the data while it's translated. The first construction of fully homomorphic encryption fashion was grounded on the proposition of algebraic chassis on the time

2009. SFHE schemes are impracticable and hamstrung because of the huge difference between the computational complications of recycling the cipher textbook and the corresponding plaintext. The major donation to this high complication is by large communication expansion due to the larger public key size. It's observed that the outturn, quiescence, and security position isn't sufficient for rearmost dispatches, particularly biomedical signals and images.

4 Proposed Work on Hybrid Security System

Data privacy is the main intention of the suggested work. In conventional methods, data is not been saved in encoded format on the cloud as it is essential for the entire process of decryption. Generally, users of homomorphic encryption can perform operations on encrypted information. However, the proposed work mainly concentrates on homomorphic encryption utilizing the ECC approach. Due to the perceived reduced key size in the suggested homomorphic encryption technique, the size of the code text is significantly reduced and maintains the same level of confidentiality as that of RSA. The drawback of Semi Morphic Cryptography is overcome by the suggested homomorphic encryption strategy that is based on ECC by significantly decreasing the transmission as well as computation overhead. Communication, security, as well as energy consumption, are some of the benefits of the suggested technique. The proposed method combines multiplicative homomorphic encryption computations with binary data splitting [12].

The suggested technique is an initiative to present a novel technique for complicated encoding and decoding data that is based on equal programming, such that the proposed mechanism can use several centered computers to complete the tasks faster and with a higher degree of security [13].

Encryption (encoding): Data is converted into a form that is inaccessible to everyone except the authorized user in this procedure.

Decryption (decoding): The encoded data is converted back into plaintext so that the encrypted information can be deciphered in this step.

Splitting: The key aspect of the methodology used in this research work for cloud computing is the splitting procedure. The splitting method splits files into more or less than two sections that are not directly related to one another but must be accessed by the only data owner. To obtain the actual information, the separated parts of the file must be merged.

In most, modern cryptography, the ability to retain encoded data safe as well as confidential is centered on a number called a key that should be used with the cryptographic process to provide an encrypted outcome or to decipher the encrypted message, rather than the crypto graphical computation itself. It is simple to decode with the correct key. When you do not have an appropriate key, then decoding becomes difficult and can be tedious sometimes. The application of encrypting as well as decrypting keys is illustrated in this proposed work [14].

- Public-Key-Encryption
- Symmetric-Key-Encryption
- Key-Length as well as Encryption-Strength

4.1 Proposed Architecture of Hybrid Cryptography System for High Security and Cloud-Based Storage

Data privacy, as well as data sharing security, are the main objectives of the suggested work. As illustrated in Figure 1, the suggested design executes the following functions:

- Begin the implementation procedure.
- Obtain the information to upload over the cloud
- Data encryption is carried out by employing a hybrid cryptography algorithm that combines AES as well as ECC.
- Upload the encoded file to a cloud storage database using the owner's private key.
- Consider a binary representation of the data file that is stored in the user account is sent to the specified receiver and thus uploading it to cloud storage.
- Obtain the binary bits of data and then combine them by employing the split keys.
- Obtain the encrypted information by combining i.e., recombination of split files.
- Hybrid cryptography to decrypt the data is employed and then the private key is exchanged with only the authorized users to restore actual information.
- End the procedure.

Once cryptography is performed, the file containing the secret data is divided into "k" numbers as well as saved in the database chosen at random. Such data-sharing methods are capable of converting ciphering datasets into a binary '0' as well as '1' format.

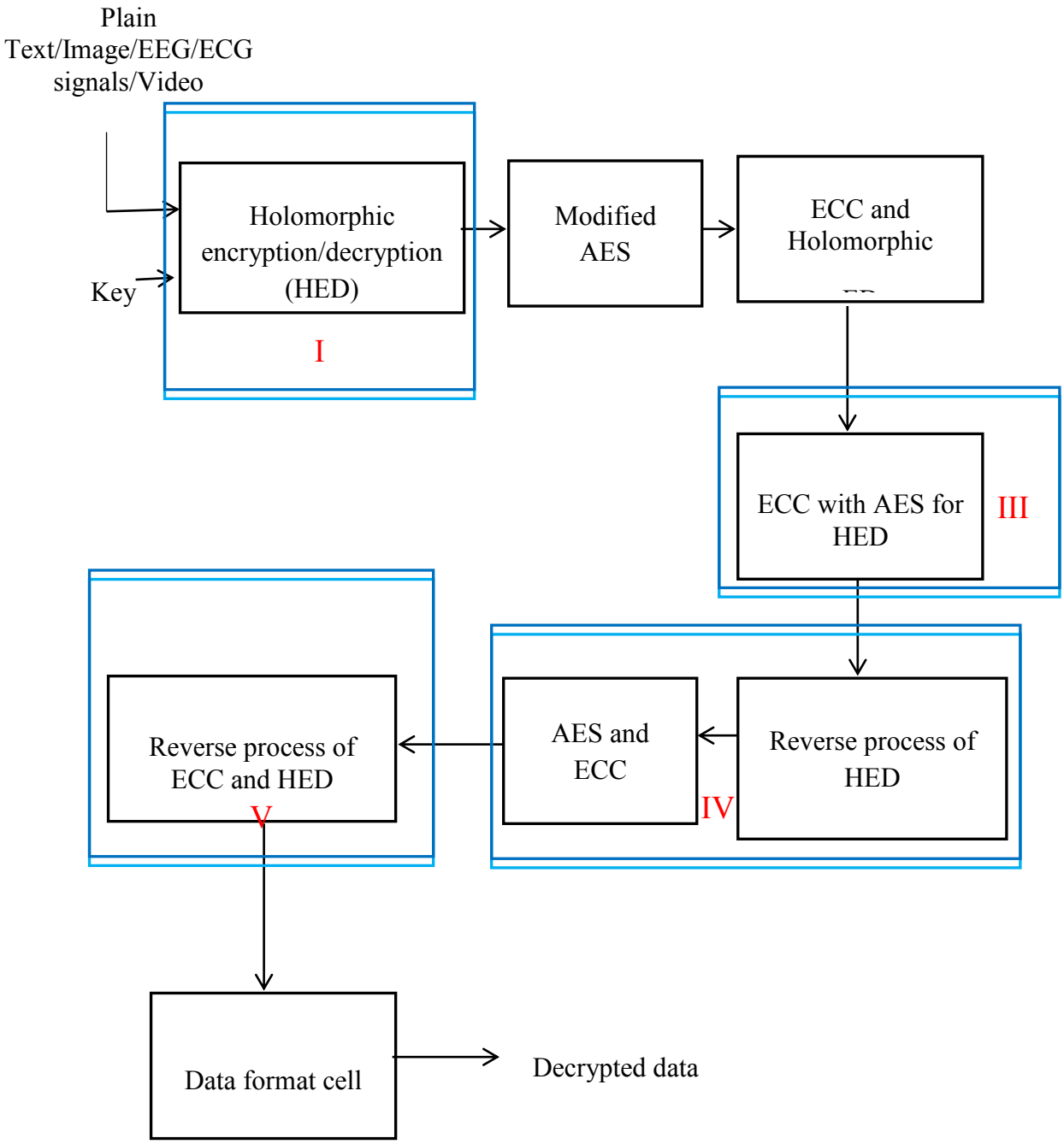

Fig. 1: Proposed overall block diagram of Multi data encryption and decryption system

Further, the ciphered data is divided among a group of private users, each of which receives one part of the split data. The suggested cryptographic algorithm is thus used to evaluate the method of sharing split data among all groups of users. All the users must be authorized to duplicate the split data, as well as the combination of all portions of the split data intended to recreate the split data. Once the information has been restored, the data must be decrypted from its binary form to its actual state [15].

Problem: A private data D should be divided into various parts so that data of "k" bits is needed to restore the hidden data D.

Algorithm: Splitting as well as combining Files. The suggested approach divides the actual information into distinct parts, encodes it, and then distributes it across multiple clouds. The separated information is then converted back into source information and decoded by employing the keys. For exploring the file contents, metadata is necessary. The following procedures are being used to split as well as combine the data.

- Start a process by uploading the file to the cloud
- Upload a document by a user name: Monitor
  For example, uploaded document: Test.doc
  Validate the document's name as well as the private keys.
  For example, Name of document: Test.doc
  Password: 12345
  Using the private key, generate a unique random number that will be used as the key
  For example, Secret key: Tes12
  Divide the document and also the key into two halves, each with a binary value of 0 as well as 1.
  Splitting of File as follows
  Name: Monitor0, Monitor1
  Secret keys: Tes1, Tes2 respectively
- Use the split parts of the keys to encrypt the split parts of the document.
  Encryption of files is completed as well as stored in the cloud. To acquire the document, recombine the splits. Merge the splits sections of the document via their corresponding split keys to download the document. Finally, with the help of a private key, decrypt the files and then merge them into the original text.
  End

5 Simulated Results of Proposed Cryptography System

According to the suggested model, the testing configuration of every user is divided into three cloud network modules. The suggested work's simulation structure is based on the development of a dynamic network employing MATLAB 2017a as well as the cloud server thingspeack.com.

We combine the technique of splitting as well as cryptography to maximize and obtain enhanced protection in data sharing following the procedure described in this suggested approach. The desired outcomes of the suggested method were determined using PyCharm 2021.2.1 with WampServer, a parallel event-driven test system. The application includes a very well structure and the modules are listed as follows:

Module 1

- From the client machine, the user sends an access request to the cloud server. With the help of a cloud network, a connection is formed between the receiver as well as the sender.

Module-2

- Login- The user, as well as the admin, will be assisted by the login page in logging into the system using a username as well as a password that has been accepted by the server.
- Selection of Key- Using data encryption as well as data decryption, the key is selected based on the user signed into the system
- Uploading- Next, the user will choose the document that must be uploaded into the system with the appropriate username and password.
- Computations – Calculations are executed based on the file chosen, and the outputs obtained after execution are sent to encryption components.
- The user information or system-executed information is encoded as well as saved in the cloud.

Module-3

- **Splitting**: The user's file will be divided into 2 halves in this phase using the binary representation.
- **Decrypt**- The encoded file will be decoded by the user through a hidden or secret key provided by the server as well as shared with both senders as well as the recipient.
- **Clubbing**: After the document has been encrypted, the split file will be merged with the original data in this step.
- **Retrieve**- This phase retrieves the data requested by the user from the cloud storage and exhibits it at the customer end.

Point Addition and Point Doubling on F_p:

In the point addition, addition is achieved over two points on the elliptic curve with various x coordinates. Let us assume those two points be A (x_A, y_A) and B (x_B, y_B) on the EC ($A \neq B$), a line is drawn from point A to point B. This line, when extended, intersects the elliptic curve at a third point –C. Then the point –C is reproduced in the x-direction for acquiring point C as represented in Figure. 4.5(a & b) which indicates the resulting point addition operation $A + B = C$. If $A = -B$, then by extending a line by joining the points A and B provide a vertical line which is extended to meet the point at infinity. On the other side (Figure 4.6). Point Doubling operation is the recurring method of adding point A to itself (i.e., $A + A = 2A = C$). This operation is performed by drawing a tangential line to EC at point A and which meets the elliptic curve at point C. By reproducing the point at C in x-direction point C can be attained.

Let p=(x_1, y_1) and q=(x2, y_2), both p & q are belongs to ff_p then p+q=(x3, y_3)
Where

$$a_3 = \left[-a_1 - a_2 + \left(\frac{b_2-b_1}{a_2-a_1}\right)^2\right] mod\ p \quad (4.6)$$

$$b_3 = \left[-b_1 + \left(\frac{b_2-b_1}{a_2-a_1}\right)(a_1 - a_3)\right]\ mod\ p \text{---- (a)}$$

To add the 2 points using point addition, for example P = (16, 8) and Q =(19, 28)

$\lambda = (b_2-b_1)/(a_2-a_1) = 20/3 = 20*21 \bmod 31 = 17$

$\lambda = 17$

$a_3 = \lambda^2 - a_1 - a_2 = 17^2 - 19 - 16 = 254 \bmod 31 = 6$

$b_3 = \lambda(a_1 - a_3) - y_1 = 17(16-6) - 8 = 162 \bmod 31 = 7$

Hence P+Q = (a_3, b_3) i.e. (16, 8) + (19, 28) = (6, 7)
i.e. 9P +18P = 27P.

To add the 2 points using point doubling P = (18,29)

$\lambda = (3a_1^2+p_1)/(2b_1) = (3*18^2+1)/(2*29) = 12/27 = 12*23 \bmod 31 = 28$

$\lambda = 28$

$a_3 = \lambda^2 - a_1 - a_2 = 28^2 - 18 - 18 = 748 \bmod 31 = 4$

$b_3 = \lambda(a_1 - a_2) - b_1 = 28(18 - 4) - 29 = 363 \bmod 31 = 22$

Hence 2(18, 29) = (4, 22)
i.e. 2P +2P = 4P.

Because asymmetric key cryptography is required for communication, the application's performance suffers significantly. The fundamental disadvantage of most public-key cryptography techniques is that they need complex mathematical computations. As a result, efficient implementations of the methods are possible. There are primarily two techniques for developing efficient cryptography systems. The first and most important scheme in software platforms focuses on executing and improving cryptographic algorithms. The advantages of this approach are that it is inexpensive and does not require any additional gear. The benefits of this technique, on the other hand, are limited by the microprocessor's design constraints. Computations on huge numbers are not performed as efficiently on microprocessors as they can on bespoke hardware. Also, the software can easily be manipulated and therefore providing security for the application. Though software executions are customized to utilize the processor's architecture [36–37] they do not match the given hardware implementations. Hardware's inherent parallelism, flexibility, and standard design significantly speed up the implementation. Hardware devices, unlike software, can be easily modified. For cryptography purposes, this is useful. Software, on the other hand, is less expensive than hardware and has a limited number of resources. Hardware design techniques are more sophisticated, and memory is another constraint for such schemes. As a result, compact, scalable, and integrated hardware solutions that are tailored to the specific application are required. FPGAs are hardware platforms that can be reconfigured. The uses of both hardware and software platforms are discussed. Also, they present more programmability and software platforms of less cost, similarly, hardware employment affords better performance than a software implementation. Meanwhile, in affine coordinates, one inversion operation is required for each addition or doubling, and the number of inversions required increases as the number of point additions or doublings increases. This disadvantage is overcome by using projective coordinates, however, this comes at the cost of more memory, as Z-coordinates for all points being processed must be saved.

Point doubling (PD): If input is single point then Point Doubling can be employed to generate point doubling, for example 2p=p+p, 4p= 2p+2p, 8p=4p+4p, etc. To implement PD on a single point say p,2p,4p…, so on, let us consider p=(a_1,b_1) then 2p=(a_3,b_3),

where $x_3 = \left[-2a_1 + \frac{3a_1-G1}{2b_1^2}\right] mod\ p$ and $y_3 = \left[-b_1 + \frac{3a_1-G1}{2b_1^2}\right] mod\ p$

256 focal points are formed using Point Addition and Point Doubling, and the same focal points are listed in Table.1. Furthermore, the Table is made up of 16x16 frames. Each point allows for the movement of x and y. The x and y coordinates are blended into a single by inserting two special characters between them to facilitate smooth operation through advanced processors. Each coordinate is 8 bits in size, and a special character is 8 bits in size; the overall size of the point is 32 bits, as shown in equation (1). Key $_{special\ characters}$= (70&*86)= (x &* y) --(1)

The binary representation of 70 is 1000110, the binary representation of 86 is 1010110, the binary representation of and is 00100110 and the binary representation of * is 00101010, therefore equation (1) can be written as Key$_{special\ characters}$=10001101010110001001100010101O.

32-bit bit of double data is input to the enciphertexting. After successful calculations of 256 focal points using point addition and point doubling, every focal point is stored in Look-Up- Table (LUT) and depends upon the 8- bit input data; 32 bits of LUT value are named. The 32- bit LUT regard is transmitted through dispatch subsystems similar to Routers or Network Interface (NI). To reduce power

dispersions in NI before transmitting the data via a communication channel, also data encryption is performed; the attained decoded information provides lower security. In the proposed work, we present three approaches to reducing the number of transformations; each approach is an advanced version of the previous method. The three approaches are anatomized regarding power dispersion which is associated with numerous variations. Encryption is a process of converting the original information into integer information to enhance the security position in this exploration work. The ECC with Elgamal is a largely-developed system for more effective and reduced fine operations in pall security, which provides protection depending on the severity of colorful problems. This system put up corresponding security with limited expenditure. The characteristics of this designed algorithm are, that it fully depends on the key and in order, values are depicted in Table.1. Likewise, it provides an advanced response for the original information, and it enables the transmission of keys securely between the sender and receivers. In general, the ECC-Holomorphic-Elgamal algorithm carries out computation operations similar to Point doubling and point addition on the information and, translated data performs some addition and modulo operations. Also, it's the most suitable result for guarding the original information security issues in the pall. The important benefits of these ultramodern systems are rapid-fire computation and security improvement. Hence, the ECCHE system is realized in this research work to confidentially allocate the cloud data. In this work, the private Key exceptional characters are shown in eq (1) and input information i.e. EEG/ECG and Image are applied as input and are depicted in Fig.4.3, latter the typical public key from the Table.1. is generated and reliant on the author's work in the ECC curve. Afterward, by using the encryption process the Ciphertext is created, and the information is encrypted via Elgamal, generated public key and point on the ECC curve. Later, the Holomorphic process is given to the encrypted information which is reliant on point doubling and points addition property. Furthermore, three constants, for example, each encrypted information x, y, and z are processed, finally, the encrypted information is passed on and uploaded to the cloud, and uploaded information is in the form of .csv and interlaced as a graph and is represented in Fig.4.4.

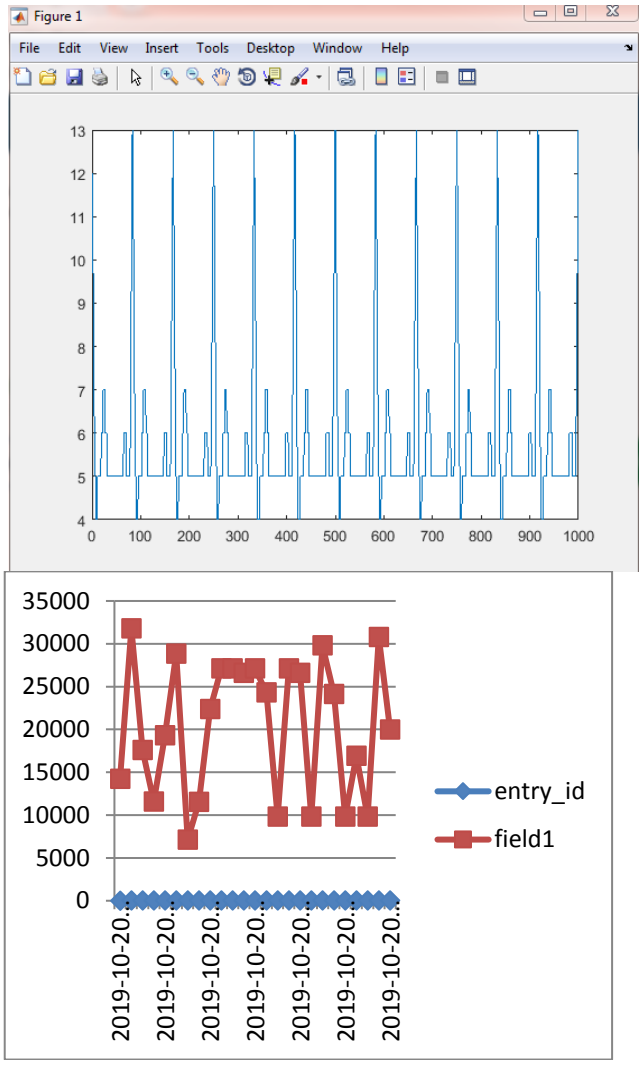

Fig.4.3 ECG representation

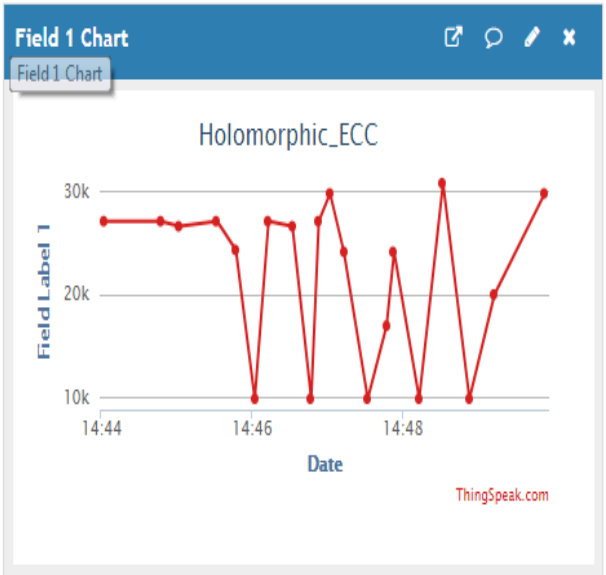

Fig. 2: Encrypted data storage in the Cloud in form of a graph

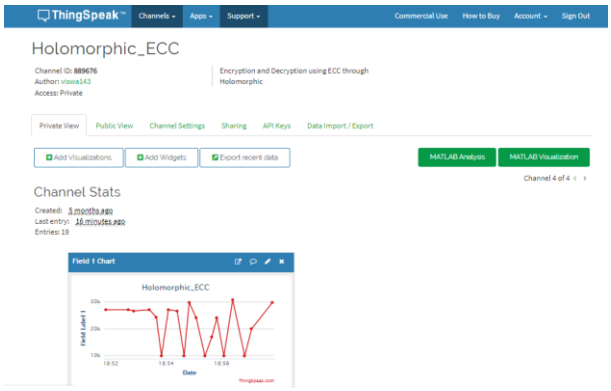

Fig. 3: Cloud Storage and its graphical representation of data

Table 1. Comparison between the existing and proposed system

Parameter	Proposed	Existing
Execution time	178ms	271ms[3]
Throughput	0.45 GHz	0.21GHz[3]
Packet delivery rate	402MHz	371MHz[3]

7 Conclusion

AES, as well as RSA, are the two primary algorithms that are employed for generating keys via cryptography in cloud computation. However, the tasks involved in cloud computing security require symmetric algorithms to encompass functions such as scanning for metadata in the decrypted information. Both contents, as well as the user's privacy, are secured using these two algorithms. When it comes to data storage, data security plays a significant role in cloud computing. There exist many techniques for cloud security. However, this research introduces a unique hybrid method by integrating AES as well as ECC techniques along with the splitting algorithm. The suggested hybrid approach offers enhanced security over data sharing as well as cloud storage attacks. We can divide any document into binary forms by using the hybrid algorithm. We employ security keys to accomplish the encryption-decryption procedure upon splitting. The proposed new fashion for cryptography is more doable until protection, velocity, and power is complex as the structure employs only XOR and simple experimental functions which use many coffers. The bit process for both enciphering and deciphering is reduced, in some of the operations analogous to Network operation Centre and Security operation Centre chips, power is used for the functioning of wired and distant communication used for rapid-fire-fire switching action among any two- information gathering and communication. ECCHE for decoding and picture are primarily reduced, power consumption and ECCHE are combined with ECC which are used for exceptional secret keys to encipher and decipher the constant data via wired or remote channels. The proposed design is executed using MATLAB 2017a. To specify the limitations of the former work a new system is designed for guaranteeing the security and protection of the information maintained in the pall. This structure consists of three modules, access policy control, encryption, and decoding. Also, it comprises rudiments analogous to the information director (council director), CSP, and information client. Firstly, the information user encrypts the information by using the ECCHE algorithm, and the pall garçon stores the information.

References:

[1]. Cheon, J.H., Kim, A., Kim, M., Song, Y. (2017). Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Takagi, T., Peyrin, T. (eds) Advances in Cryptology – ASIACRYPT 2017. ASIACRYPT 2017. Lecture Notes in Computer Science(), vol 10624. Springer, Cham. https://doi.org/10.1007/978-3-319-70694-8_15.

[2]. Peter, A., Kronberg, M., Trei, W., Katzenbeisser, S. (2012). Additively Homomorphic Encryption with a Double Decryption Mechanism, Revisited. In: Gollmann, D., Freiling, F.C. (eds) Information Security. ISC 2012. Lecture Notes in Computer Science, vol 7483. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-33383-5_15.

[3]. Li, N. (2018). Homomorphic Encryption. In: Liu, L., Özsu, M.T. (eds) Encyclopedia of Database Systems. Springer, New York, NY. 2018. https://doi.org/10.1007/978-1-4614-8265-9_1486

[4]. Munjal, K., Bhatia, R. A systematic review of homomorphic encryption and its contributions in healthcare industry. *Complex Intell. Syst.* (2022). https://doi.org/10.1007/s40747-022-00756-z

[5]. Stehlé, D., Steinfeld, R. (2010). Faster Fully Homomorphic Encryption. In: Abe, M. (eds) Advances in Cryptology - ASIACRYPT 2010. ASIACRYPT 2010. Lecture Notes in Computer Science, vol 6477. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-17373-8_22

[6]. Cheon, J.H., Han, K., Kim, A., Kim, M., Song, Y. (2018). Bootstrapping for Approximate Homomorphic Encryption. In: Nielsen, J., Rijmen, V. (eds) Advances in Cryptology – EUROCRYPT 2018. EUROCRYPT 2018. Lecture Notes in Computer Science(), vol 10820. Springer, Cham. https://doi.org/10.1007/978-3-319-78381-9_14

[7]. Y. Yoon and J. Moon, "Verifying the Integrity of Private Transaction Information in Smart Contract using Homomorphic Encryption," 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), 2019, pp. 38-40, doi: 10.1109/ECICE47484.2019.8942648.

[8]. Pan Yang et.al, "An Efficient Secret Key Homomorphic Encryption Used in Image Processing Service", Security and Communication Networks, Hindawi, Volume 2017, Article ID 7695751, 11 pages, https://doi.org/10.1155/2017/7695751

[9]. J. Basilakis and B. Javadi, "Efficient Parallel Binary Operations on Homomorphic Encrypted Real Numbers," in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 1, pp. 507-519, 1 Jan.-March 2021, doi: 10.1109/TETC.2019.2906047.

[10]. A. C. Mert, E. Öztürk and E. Savaş, "Design and Implementation of Encryption/Decryption Architectures for BFV Homomorphic Encryption Scheme," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 2, pp. 353-362, Feb. 2020, doi: 10.1109/TVLSI.2019.2943127.

[11]. A. Al Badawi, Y. Polyakov, K. M. M. Aung, B. Veeravalli and K. Rohloff, "Implementation and Performance Evaluation of RNS Variants of the BFV Homomorphic Encryption Scheme," in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 2, pp. 941-956, 1 April-June 2021, doi: 10.1109/TETC.2019.2902799.

[12]. Y. Ke, M. -Q. Zhang, J. Liu, T. -T. Su and X. -Y. Yang, "Fully Homomorphic Encryption Encapsulated Difference Expansion for Reversible Data Hiding in Encrypted Domain," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 8, pp. 2353-2365, Aug. 2020, doi: 10.1109/TCSVT.2019.2963393.

[13]. Z. Zhang, P. Cheng, J. Wu and J. Chen, "Secure State Estimation Using Hybrid Homomorphic Encryption Scheme," in IEEE Transactions on Control Systems Technology, vol. 29, no. 4, pp. 1704-1720, July 2021, doi: 10.1109/TCST.2020.3019501.

[14]. Xun Wang. et.al, "A More Efficient Fully Homomorphic Encryption SchemeBased on GSW and DM Schemes", Hindawi, Security and Communication Networks, Volume 2018, Article ID 8706940, 14 pages, https://doi.org/10.1155/2018/8706940.

[15]. Ambika Pawar, Ajay Dani, A Novel Approach for Protecting Privacy in Cloud Storage based Database Applications, WSEAS TRANSACTIONS on COMPUTERS,E-ISSN: 2224-2872, Volume 15, 2016

[16]. S. K. Kermanshahi et al., "Multi-Client Cloud-Based Symmetric Searchable Encryption," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 5, pp. 2419-2437, 1 Sept.-Oct. 2021, doi: 10.1109/TDSC.2019.2950934.

[17]. M. M. Panchbhai and U. S. Ghodeswar, "Implementation of point addition & point doubling for Elliptic Curve," 2015 International Conference on Communications and Signal Processing (ICCSP), 2015, pp. 0746-0749, doi: 10.1109/ICCSP.2015.7322589.

[18]. T. Chen, H. Li, K. Wu and F. Yu, "Countermeasure of ECC against Side-Channel Attacks: Balanced Point Addition and Point Doubling Operation Procedure," 2009 Asia-Pacific Conference on Information Processing, 2009, pp. 465-469, doi: 10.1109/APCIP.2009.250.

[19]. W. Wang, Y. Hu, L. Chen, X. Huang, and B. Sunar, "Exploring the feasibility of fully homomorphic encryption," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 698–706, Mar. 2015

[20]. Y. Li and L. Xiao, "Parallel DNA Computing Model of Point-Doubling in Conic Curves Cryptosystem over Finite Field GF(2^n)," 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2019, pp. 1564-1571, doi: 10.1109/HPCC/SmartCity/DSS.2019.00215.

[21]. C. Andres Lara-Nino, A. Diaz-Perez and M. Morales-Sandoval, "A comparison of Differential Addition and Doubling in Binary Edwards Curves for Elliptic Curve Cryptography," 2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), 2021, pp. 12-18, doi: 10.1109/WorldS451998.2021.9514047.

[22]. A. Sakthivel and R. Nedunchezhian, "Decreasing point multiplication over ECC (Zp) using tree computations," 2012 International Conference on Computing, Communication and Applications, 2012, pp. 1-5, doi: 10.1109/ICCCA.2012.6179229.

[23]. M. M. Islam, M. S. Hossain, M. K. Hasan, M. Shahjalal and Y. M. Jang, "FPGA Implementation of High-Speed Area-Efficient Processor for Elliptic Curve Point Multiplication Over Prime Field," in IEEE Access, vol. 7, pp. 178811-178826, 2019, doi: 10.1109/ACCESS.2019.2958491.

[24]. M. R. Hossain, M. S. Hossain and Y. Kong, "Efficient FPGA Implementation of Unified Point Operation for Twisted Edward Curve Cryptography," 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), 2019, pp. 1-4, doi: 10.1109/IC4ME247184.2019.9036635.

[25]. N. Pirotte, J. Vliegen, L. Batina and N. Mentens, "Design of a Fully Balanced ASIC Coprocessor

Implementing Complete Addition Formulas on Weierstrass Elliptic Curves," 2018 21st Euromicro Conference on Digital System Design (DSD), 2018, pp. 545-552, doi: 10.1109/DSD.2018.00095.

[26]. M. A. Mehrabi, C. Doche and A. Jolfaei, "Elliptic Curve Cryptography Point Multiplication Core for Hardware Security Module," in IEEE Transactions on Computers, vol. 69, no. 11, pp. 1707-1718, 1 Nov. 2020, doi: 10.1109/TC.2020.3013266.

[27]. Qingwei Li, Zhongfeng Wang and Xingcheng Liu, "Fast point operation architecture for Elliptic Curve Cryptography," APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems, 2008, pp. 184-188, doi: 10.1109/APCCAS.2008.4745991.

[28]. M. Shirase, "An Improved Addition Formula on Elliptic Curves Given by Weierstrass Normal Form," 2013 16th International Conference on Network-Based Information Systems, 2013, pp. 528-533, doi: 10.1109/NBiS.2013.88.

[29]. D. M. Schinianakis, A. P. Fournaris, H. E. Michail, A. P. Kakarountas and T. Stouraitis, "An RNS Implementation of an F_{p} Elliptic Curve Point Multiplier," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 56, no. 6, pp. 1202-1213, June 2009, doi: 10.1109/TCSI.2008.2008507.

[30]. L. Marin, "Differential Elliptic Point Addition in Twisted Edwards Curves," 2013 27th International Conference on Advanced Information Networking and Applications Workshops, 2013, pp. 1337-1342, doi: 10.1109/WAINA.2013.152.

[31]. I. Kabin, Z. Dyka, D. Klann, N. Mentens, L. Batina and P. Langendoerfer, "Breaking a fully Balanced ASIC Coprocessor Implementing Complete Addition Formulas on Weierstrass Elliptic Curves," 2020 23rd Euromicro Conference on Digital System Design (DSD), 2020, pp. 270-276, doi: 10.1109/DSD51259.2020.00051.

[32]. Meloni, N. (2007). New Point Addition Formulae for ECC Applications. In: Carlet, C., Sunar, B. (eds) Arithmetic of Finite Fields. WAIFI 2007. Lecture Notes in Computer Science, vol 4547. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-73074-3_15

[33]. Joye, M. (2008). Fast Point Multiplication on Elliptic Curves without Precomputation. In: von zur Gathen, J., Imaña, J.L., Koç, Ç.K. (eds) Arithmetic of Finite Fields. WAIFI 2008. Lecture Notes in Computer Science, vol 5130. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69499-1_4

[34]. K. Phalakarn, K. Phalakarn and V. Suppakitpaisarn, "Optimal Representation for Right-to-Left Parallel Scalar Point Multiplication," 2017 Fifth International Symposium on Computing and Networking (CANDAR), 2017, pp. 482-488, doi: 10.1109/CANDAR.2017.14.

[35]. G. Yang, F. Kong and Q. Xu, "Optimized FPGA Implementation of Elliptic Curve Cryptosystem over Prime Fields," 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2020, pp. 243-249, doi: 10.1109/TrustCom50675.2020.00043.

[36]. P. Das, D. B. Roy, H. Boyapally and D Mukhopadhyay, "Inner collisions in ECC: Vulnerabilities of complete addition formulas for NIST curves," 2016 IEEE Asian Hardware-Oriented Security and Trust (AsianHOST), 2016, pp. 1-6, doi: 10.1109/AsianHOST.2016.7835562.