JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 10, Issue.12, December 2022 E-ISSN: 2347-2693

Enhacing Cloud Security: Combining Homomorphic and Elliptic Curve Cryptography for Resilient Fusion

Madhira Srinivas[1*], Porika Sammulal[2]

[1,2]Department of Computer Science & Engineering, JNTUH University, Hyderabad, Telangana, India

**Corresponding Author: madhirasri@gmail.com*

DOI: https://doi.org/10.26438/ijcse/v10i12.4146 | Available online at: www.ijcseonline.org

Received: 27/Oct/2022, Accepted: 28/Nov/2022, Published: 31/Dec/2022

Abstract **—** This study introduces a unique method for strengthening healthcare data security by combining homomorphic encryption (HE) with elliptic curve cryptography (ECC) in a way that is both robust and effective. Patients' right to privacy and data security must be protected as the cloud computing industry moves towards a model based on data-driven insights. For safe data processing and analytics, we provide a hybrid system that takes advantage of ECC's efficiency in key exchange while also including HE's privacy-preserving characteristics. We illustrate the efficacy of our suggested strategy through a thorough comparative analysis, actual implementation, and case study in a Cloud computing setting. The hybrid architecture prevents sensitive information from falling into the wrong hands and facilitates secure and efficient data flow among legitimate parties. Our results highlight the promise of this robust union to revolutionize cloud securityand execution time,encryption,and decryption time compared to other crypto algorithms, opening doors to betterand more efficient programs for future endeavors.

Keywords **—**Homomorphic encryption, Elliptical curve cryptography, Hybrid Framework, Cloud Computing, Comparative Analysis.

I. INTRODUCTION

Everything from the size of our computers to the speed of our computers to the size of our discs to the efficiency of our networks to the bandwidths we enjoy today rose enormously. Enterprise cloud computing is not only an academic term but also a commercial concept [1], [2], and [3]. It consists of infrastructure and hundreds of applications. The term "cloud storage" refers to a relatively new model of data storage that evolved out of cloud computing; in this model, a hosting provider is responsible for the hardware infrastructure, and users' data is stored in logical pools across multiple servers. But people are concerned about privacy and security breaches in data storage. As a consequence, it is challenging to promote and gain widespread adoption of cloud storage services due to concerns about data security. With the rise of cloud computing, safeguarding personal data in cloud storage has emerged as a critical concern. Numerous studies have been conducted recently on privacy protection strategies for cloud storage. Here we take a look at the most up-to-date research on cloud storage with regard to the storage of cipher, auditing of security, and controlling of cipher access[5]. Important advances in homomorphic encryption [6-10].

For the first time, Rivest, Adleman, and Dertouzos[11] presented whether a third party could compute the data using just the cipher text and not the key. The calculated results are then provided to the user. Years of labor eventually paid off when cryptography pioneer Gentry developed the first comprehensive encryption method based on an ideal lattice, which sped up the investigation into a fully homomorphic encryption scheme [12]. There are, however, issues with the current architecture of the completely homomorphic encryption method, such as a public key complex and heavy programming, an expansion rate of the cipher text that is too big, and a computation time that is too long. Both the RSA algorithm and its equivalent operation F are homomorphisms for multiplication [13], whereas the Paillier method is a homomorphism for addition [14]. Therefore, the technique cannot be used in practice; instead, we suggest a homomorphic encryption strategy based on Elliptical curve arithmeticcryptography (ECC).

As of the end of 2015, the Seismological Bureau of Fujian Province has 61 GPS base stations. We actively monitor crustal movement in the Fujian province every day by calculating real-time GPS data. This requires a large number of calculations to accomplish. However, GPS data is classified information, necessitating encryption before calculation if the problem is typically tackled using cloud computing. Since elliptic curve cryptography (ECC) has greater security and less complexity, this work proposes a method for homomorphic encryption with privacy protection using ECC [15]. And we use it to calculate earthquake GPS data, ensuring accurate calculations and safe storage of GPS information.

.

II. BACKGROUND AND RELATED WORK

Data security, encryption methods, and decryption strategies are all topics that will be covered in the works presented here.

When it comes to cloud storage, there are [16] distinct methods of protecting personal information using homomorphic encryption. When using homomorphic encryption, the user may do computations on the encrypted data, and the decrypted output will be identical to the original. Partial Homomorphic Encryption (PHE) and Fully Homomorphic Encryption (FHE) are the two varieties of homomorphic encryption. Unlike completely homomorphic, where both addition and multiplication are possible, PHE only supports the former. To ensure the safety of information stored in the cloud, [17] created a Hybrid Data Encryption Method. Here For encryption and decryption, we employ Blowfish and RSA, and for authentication, we use a digital signature. Xilinx ISE 14.1 is used for the implementation. Both symmetric and asymmetric algorithms may use this method. Blowfish is an open-source algorithm, hence it is quite cost-effective.

Homomorphic encryption's applicability to hybrid systems is explored in [18]. Homomorphic encryption has several applications and is utilised in e-voting, healthcare, and other sectors. Because perfect homomorphic encryption is still under development, we may mask its shortcomings by including homomorphic encryption in hybrid cryptosystems.Using homomorphic encryption, [19] created a new hybrid encryption standard. To perform homomorphic addition, the authors of this study used the Paillier technique, whereas RSA encryption was utilized to do homomorphic multiplication. The cipher text is saved in the public cloud, while the encryption procedure is done on a private cloud.

Homomorphic encryption for network encoding was developed in [20]. Homomorphic signatures may be used to protect network encoding against tampering. Recently, he has developed a homomorphic signature plot based on the RSA algorithm for this same purpose. We show that in practice, their scheme does not satisfy the necessary homomorphic quality and that even if it is resolved efficiently, no message-forging attacks are possible. Attribute-based encryption for use in a hybrid cloud environment was developed in [21]. Here, he designed the most consolidated frameworks that provide data access to a cloud user who satisfies a set of criteria. One method of challenging such rules is to store and manage user data in an authorized cloud server. When the security of just one of the servers storing information is breached, the security of all user information is breached. Information owners must maintain attribute-based security to encode the stored information in order to gain control over access, keep data secure, and receive accurate computation results. Cloud servers might be compromised by forged ciphertext during data delegation. Furthermore, responding that they are not authorized might confuse the authorized users.

In most cases, the encryption key determines the randomness of the attribute policies. In this work, we introduce Cipher-text Policy Attribute-Based Encryption, an approach to secure data encryption that allows for flexible, auditable permission based on cipher-text policies. Whether or if the capacity server is compromised, the encoded data remains private thanks to the suggested method. In addition, we have extensive proof that our technique can withstand collision attacks. The assessment of the suggested framework's execution is presented in advance, along with an explanation of how the equivalent would be implemented.

Improve cloud security using the modified Elliptic curve technique proposed in [22]. An identical MECC computation is used for encoding and decoding the data. Here, customers and other administrators are verified through authentication in order to access the cloud data they need. Once the requesters have been verified, they will be assigned characteristics. The data collectors are the ones responsible for doing the MECC computation and generating the private key needed to decode the encrypted data. In cloud computing terms, this ensures the highest possible data epitome. We compared the suggested plan's execution to that of more conventional plans and found that the MECC computation provides much more security. ElGamal elliptic curve cryptosystem for distributed computing in the cloud was created in [23]. Customers may entrust their sensitive data to the cloud without worrying about security thanks to the use of server-side cryptosystems for encryption and decryption. In this article, the author proposes yet another ElGamal Elliptic Curve cryptosystem calculation assisted by cloud servers. Clients of connected cloud services will benefit from the proposed standards since they can thwart both active and passive threats.

In most cases, the encryption key determines the randomness of the attribute policies. In this work, we introduce Cipher-text Policy Attribute-Based Encryption, an approach to secure data encryption that allows for flexible, auditable permission based on cipher-text policies. Whether or if the capacity server is compromised, the encoded data remains private thanks to the suggested method. In addition, we have extensive proof that our technique can withstand collision attacks. The assessment of the suggested framework's execution is presented in advance, along with an explanation of how the equivalent would be implemented.

Improve cloud security using the modified Elliptic curve technique proposed in [22]. An identical MECC computation is used for encoding and decoding the data. Here, customers and other administrators are verified through authentication in order to access the cloud data they need. Once the requesters have been verified, they will be assigned characteristics. The data collectors are the ones responsible for doing the MECC computation and generating the private key needed to decode the encrypted data. In cloud computing terms, this ensures the highest

possible data epitome. We compared the suggested plan's execution to that of more conventional plans and found that the MECC computation provides much more security. ElGamal elliptic curve cryptosystem for distributed computing in the cloud was created in [23]. Customers may entrust their sensitive data to the cloud without worrying about security thanks to the use of server-side cryptosystems for encryption and decryption. In this article, the author proposes yet another ElGamal Elliptic Curve cryptosystem calculation assisted by cloud servers. Clients of connected cloud services will benefit from the proposed standards since they can thwart both active and passive threats.The Advanced Encryption Standard algorithm and Proxy Re-encryption were used to create a data storage system, as described in [26]. Here, the storage infrastructure is dispersed and lacks concentration. To provide not only safe and sound data storage and retrieval, but also the possibility for one user to communicate his data on the cloud with another user inside the encrypted arrangement itself, a proxy re-encryption conspire is suggested and combined with a distributed erasure code. This article advocates for the practice of encoding encrypted files and exchanging data in their encoded form. Both encoding and dissemination strategies are used in this study. Erasure encoding is crucial in a decentralized distributed system since it facilitates the exchange of encoded materials. The data in the cloud is verified to be secure by using a distributed erasure code.

III. TECHNIQUES USED

3.1. Homomorphic encryption

Homomorphic encryption is a method of cryptography that shifts the focus of numerical operations from the actual data to the figures representing that data. The material seen in the figures is a jumbled version of the original data (also known as plain content). It's processed and deciphered to provide the desired result. Homomorphic encryption's defining feature is that decrypting the ciphertext should result in the same output as decrypting the plaintext itself. Initial communication consists of simple text messages. In a computer system, "jane doe" may represent a real person. The point is to apply some transformation to it, such as changing the case of the name. It's doable, but it's better to encrypt the message first so that whoever processes it never sees Jane's name in plain text. This results in a garbled message with content 163726. Then, it is evaluated, or transformed, into a different value by use of some other capability. Such like... The result is another fully encrypted message with a value of 163853. By decoding this message, we may learn that "Jane Doe" is intended. Then, another function is used to evaluate it, so transforming it into a new value. The result is another fully encrypted message with a value of 163853. By decoding this message, we may learn that "Jane Doe" is intended. Homomorphic schemes appeared shortly after RSA. It is possible to conduct tasks using these schemes without decrypting the original data. To encrypt and decrypt data using a completely homomorphic encryption scheme, all that is needed are the normal "enc" and "dec" functions, respectively, plus an evaluation capability that we will refer to as "eval." This "eval" accepts the program's data and the chart's data as inputs, and produces the chart's chart message as output.

If you need a homomorphism and a computation framework, look for a homomorphic encryption plot. The instructions provide the math needed to generate both public and private keys. The public key may be used by anybody, but the private key is needed only by the person doing the deciphering. Incomplete flavors exist alongside full ones. In contrast, to complete plans, which cover all of the usual number juggling operations, half plans are designed to safeguard the structure just for particular planned tasks, such as enlargement. All of the duties here are related to the underlying mathematical structure, which for our purposes will include specific number crunching in cases when the modulus is a very large value. RSA is now utilized to produce encryption keys; these keys range in length from 1024 bits to 2048 bits and must be regularly increased. As a result, the encryption procedure incurs additional computational and storage costs.

B. Elliptical arithmetic curve Cryptography

This is an algorithm based on public-key cryptography. Its original aim was digital signatures, but it was subsequently expanded to include encryption and decryption. The method relies on a discrete logarithm to determine its efficacy.

Step 1:Pick a prime number at random “A”.
Step 2: Pickany two random numbers“B” and “C”such that b<a andc<a.
Step 3: Identify the value of the variable d
Using the formula$D = B^C \% A$.
Step 4: D is the public key and C is the private key and A and B are both public.

The randomly chosen prime number a can be found everywhere. After that, we choose two more random numbers, with b and c being the public and private ones, respectively. In akey generation, the size of the key affects both the computational time required and the level of security. This is a drawback of the key generation process.
Modern encryption methods like elliptic curve cryptography allow for both public and private encryption and decoding. It offers protection using every known algorithm, making it the most secure algorithm available at an affordable price. It also allows users to safely exchange keys with one another. It solves the cloud computing security and privacy issues well. Data stored in the cloud is encrypted using elliptic curve cryptography and homomorphic encryption. The inputs to this method are the input data and the private key. The public key is then generated by the curve's generating function g. The cipher text is similarly created with the help of four random integers rs. Next, we apply the public key to the cipher text created by the random function. The result is the encrypted information in the file.

Consider source A wants to send the encrypted plain text m to B. A should perform the following procedure to transfer the data:

1) SourceA selects the elliptic curve E and the associated base point G.

2) Source A generates a public key from the selected private key (k). K=kG.

3) Source A will send the generated variables to destination B.

4) Destination B receives this message, The plaintext is encoded to a single point M in E, and an integer r(r<n) is generated at random.

5) Destination B identifies

$C1 = M + rK \ and \ C2 = rG.$

6) Destination B will send $C1 \ and \ C2$ to user A.

7) Source A receives this message, then it calculates$C1 - kC2$ and gets a pint M.

Because $C1 - kC2 = \quad M + rk - k(rG) = M + rK - r(kG) = M$, then M is decrypted to get the plaintext.

Improved ECC

Elliptical arithmetic Curve Cryptography has a big computation complexity due to inversion operation.

Hence, we improve ECC by ignoring inversion which has a high efficiency.

1) The Hash function is notarized by the signer to create abstract data.

2) The parameter of the elliptic curve is decided by the signerF = (P, a, h, g, n, h) or (m, f(x), a, h, g, n, h).

3)The signed message is verified once the verifier receives the signed message's Hash and the elliptic curve parameter..

4) The elliptic curve point group and nite yield are used to determine the signing key x.

Then itgets public key $y = xg$ and public y.

5) Siandom number $K, 1 \leq K \leq n - 1.$

6) It computes r = kg, if r = 0 then return back step 5.

7) It identifies $s = mrx - k$ and gets (s,r) as the signature of m. (s,r) and m are sent to verier.

8) Verier calculates $r' = sg + myr.$

9) Verier judges whether n′= r, if they are equal, a signature is proper. Otherwise, it rejects the signature.

IV. PROPOSED SCHEME

4.1. Homomorphic Encryption

Homomorphic encryption allows for direct manipulation of the ciphertext without first decrypting it. In this example, we'll say that the plaintext is M=m1,m2,m3,.....mn and that the encryption function is Ek1. The decryption function is Dk2. To use this formula correctly, both the encryption and decryption functions must adhere to the Homomorphic encryption characteristic.

$$\alpha\left(E_{k1}(m_1), E_{k2}(mm_2), \ldots., E_{kn}(m_n)\right) = \beta(E_{k1}(m_1, m_2, \ldots m_n))$$

When data m_1 , $m_{2,}$ …., $m_n$conducts β operation without leaking, we can encrypt it as $\left(E_{k1}(m_1), E_{k2}(mm_2), \ldots., E_{kn}(m_n)\right),$ then do α operation for it. The result is decrypted as βm_1, $m_{2,}$ …m_n. The addition homomorphism and multiplication homeomorphism can be expressed as

$$m_1 + m_2 + \cdots + m_n = D_k\left(E_k(m_1) + E_k(m_2) + \cdots + E_k(m_{1n})\right)$$

$$m_1 . m_2 \ldots m_n = D_k\left(E_k(m_1) . E_k(m_2) \ldots . E_k(m_n)\right)$$

4.2. Modified ECC Homomorphic Encryption

We implement the homomorphisms of addition and multiplication using the enhanced ECC.

i) Homomorphic addition

Plaintext m_i is coded on one-point Pm_i in E. Randomly select a number r_i and get encrypted data $(C_i . C_i)$. It makes additive operation for $(C_{1i} . C_{2i}) \ldots (C_{in} . C_{2i})$ and obtains $(\sum_{i=1}^{n} C_{1i} . \sum_{i=1}^{n} C_{1i})$ Then calculate C= $k\sum_{i=1}^{n} C_i$. So we can prove:

$$k \sum_{i=1}^{n} C_{1i} = kG \sum_{i=1}^{n} ri = k \sum_{i=1}^{n} r_i.$$

$$\sum_{i=1}^{n} C_{2i} - C = k \sum_{i=1}^{n} r_i + \sum_{i=1}^{n} P_{mi} - k \sum_{i=1}^{n} r_i = \sum_{i=1}^{n} Pm_i.$$

So we can calluculate sum $\sum_{i=1}^{n} Pm_i$.And decrypt it to obtain $\sum_{i=1}^{n} m_i$.

ii) Homomorphic Multiplication. Plaintext m_i is calculated then it gets (C_{1i}, C_{2i} , C_{3i}). it makes multiplication operation for (C_{1i}, C_{2i} , C_{3i}) …(C_{1n}, C_{2n} , C_{3n}). Then calculate k^n.$C_{1_1}, C_{1_2} C_{1_3} \ldots \ldots C_{1_n}$ 1n through private key k. So, we canshow:

$$k^n C_{1_1} . C_{1_2} . C_{1_n} = k^n G r_1 . r_2 \ldots r_n = C_{2_1} . C_{2_2} . C_{2_n}$$

So we can get $C_{3_1} . C_{3_2} \ldots C_{3_n}$.

$$C_{2_1}^{-1} C_{2_2}^{-1} C_{2_n}^{-1} = m_1 . m_2 \ldots m_n.$$

V. EXPERIMENTAL RESULTS

Parameters like execution time, decryption time, and encryption time are compared between the proposed system and the standard technique in this section. To compare, you may look at previous work like the improved MORE algorithm and the PORE algorithm.

Execution time:

it is the amount of time it takes for data to be retrieved from a cloud service. Figure 1 displays a comparison of the times required to execute each method. The time it takes for the cloud storage server to encrypt data before storing it and decrypting it upon retrieval are also factored in. Key sizes are shown along the x-axis, while milliseconds are shown along the y-axis.

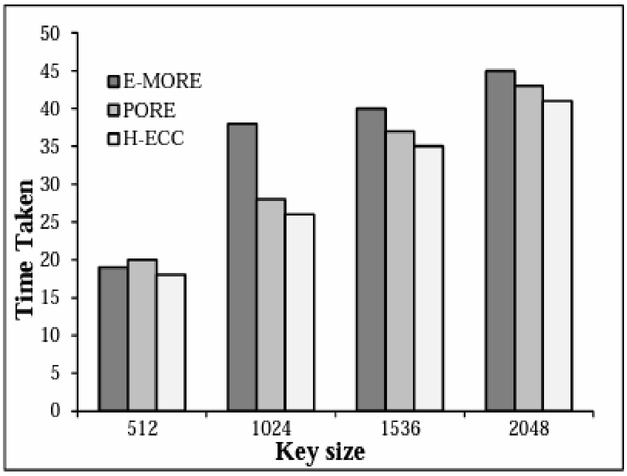

Figure 1. Execution time comparison of proposed and existing system

Encryption time:

Data encryption takes some amount of time. It may also be written as the difference between the start and finish times of an event. Figure 2 compares the proposed system's encryption time to that of many other algorithms. Several parameters, including the length of time it took to run the system and the amount of the encryption keys, are utilised in our analysis. Based on this evaluation, it seems that the suggested system outperforms the alternatives.

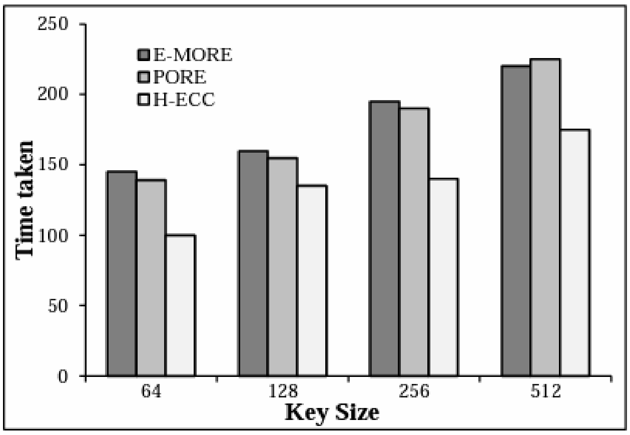

Figure 2. Encryption time comparison of proposed and existing system

Decryption Time

The length of time it takes for a certain encryption technique to decode some data. With a larger key, more time is required to unlock it. It may also be used to symbolise the beginning of the end. In this analysis, we compare many algorithms, each with their own unique set of parameters, including key size and execution time. In Fig.3, we can observe that our suggested system requires less time to decode data than the state-of-the-art approach.

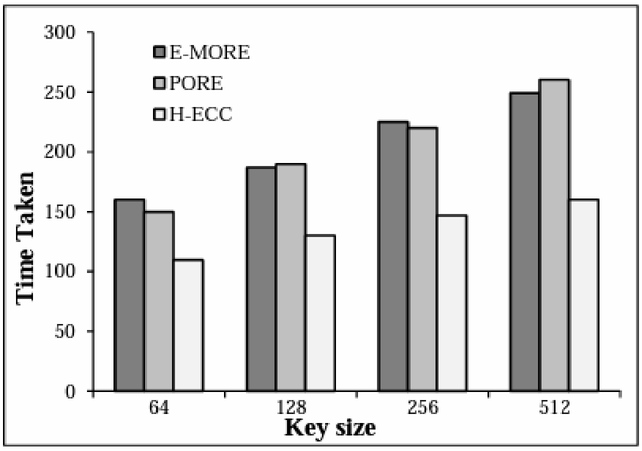

Figure 3. Decryption time comparison of proposed and existing system

VI. CONCLUSION

We proposed a new architecture that guarantees the security and privacy of the cloud environment for data storage to address the drawbacks of the prior work. Key generation, encryption, and decryption are the three fundamental components of this architecture. The information is first homomorphically encrypted and encrypted using elliptic curve cryptography. The user makes a request to the cloud service, and the service verifies, using some kind of identity verification protocol or method, whether the user is who he claims to be. Many cloud services first verify the user's identification with the use of access control policies before encrypting the data using the aforementioned technique and storing it in the cloud. The procedure of decryption is identical to that of encryption. The decryption technique is used to read the encrypted data and the private key is the central component. Many studies are being conducted to improve cloud security and privacy. In the event of a data audit, homomorphic encryption is utilised to allow for remote processing of computations. Since completely homomorphic encryption is superior than partial homomorphic encryption, it is increasingly being employed in hybrid cloud encryption processes. To protect user anonymity when searching, a fully homomorphic approach may be utilised. It's a major area of study at a lot of different businesses. Since we are simply planning to use this for testing purposes at the moment, we are not concerned about the costs associated with it. When the time comes for hardware-level implementation, we must then consider financial implications.

.

REFERENCES

[1] Fang Y C, Gao Y, Stap C "Future Enterprise Computing Looking into 2020", *Frontier and Innovation in Future Computing and Communications*, Springer Netherlands, pp.**127-134, 2014.**

[2] Zhang P, Gao Y, Fierson J, "Eigen analysis-based task mapping on parallel computers with cellular networks", *Mathematics of Computation*, Vol.**83**(288), pp.**1727-1756, 2014.**

[3] Zeng W, Zhao Y, Ou K, et al. "Research on cloud storage architecture and key technologies", *Proceedings of the 2nd*

International Conference on Interaction Sciences ICIS '09, Information Techno, pp.**1044-1048, 2009.**

[4] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz, "Maintenance-Free Global Data Storage", *IEEE Internet Computing,* Vol.**5**, Issue.**5**, pp.**40–49, 2001.**

[5] Gentry C. "A fully homomorphic encryption scheme", *Stanford, USA: Stanford University*, **2009.**

[6] Van Dijk M, Gentry C, Halevi S, et al. "Fully homomorphic encryption over the integers", *Advances in cryptology–EUROCRYPT 2010,* Springer Berlin Heidelberg, pp.**24-43, 2010.**

[7] Smart N P, Vercauteren F, "Fully homomorphic encryption with relatively small key and ciphertext sizes", *Public Key Cryptography PKC 2010,* Springer Berlin Heidelberg, pp.**420-443, 2010.**

[8] Brakerski Z, Vaikuntanathan V, "Fully homomorphic encryption from ring-LWE and security for key dependent messages", *Advances in Cryptology–CRYPTO 2011,* Springer Berlin Heidelberg, pp.**505-524, 2011.**

[9] Zhang P, Gao Y, "Matrix Multiplication on High-Density Multi-GPU Architectures: Theoretical and Experimental Investigations", *High Performance Computing. Springer International Publishing,* pp.**1-10, 2015.**

[10] Gentry C, Halevi S, "Implementing Gentry's fully-homomorphic encryption scheme", *Advances in Cryptology–EUROCRYPT 2011,* Springer Berlin Heidelberg, pp.**129-148, 2011.**

[11] V. S. Miller, "Use of Elliptic Curve in Cryptography". *In Proceedings of Advances in Cryptology (CRYPTO'85),* Springer Verlag, pp.**417-426, 1986.**

[12] GU Chun-sheng, LI Hong-wei ,et al., "CAA Attack on Privacy Preserving Computable Encryption Scheme of Cloud Computing", *Journal of Chinese Computer Systems*, Vol.**35**, Issue.**12**, pp.**2644-2649, 2014.**

[13] Zhiwei Wang, "Improvement on Ahn et al.'s RSA P-Homomorphic Signature Scheme", *Security and Privacy in Communication Networks,* Springer Berlin Heidelberg, pp.**19-28, 2012.**

[14] Penn G M, PöTzelsberger G, Rohde M, et al. "Customisation of Paillier homomorphic encryption for efficient binary biometric feature vector matching", *Biometrics Special Interest Group (BIOSIG), 2014 International Conference of the. IEEE,* pp.**1-6, 2014.**

[15] N. Koblitz, "Elliptic Curve Cryptosystems", *Mathematics of Computation*, Vol.**48**, pp.**203-209, 1987.**

[16] V.P. Bansal and S. Singh, "A Hybrid Data Encryption Technique using RSA and Blowfish for Cloud Computing on FPGAs", *Proceedings of 2nd International Conference on Recent Advances in Engineering and Computational Sciences*, pp.**103-108, 2015.**

[17] K. El Makkaoui, A. Beni-Hssane and A. Ezzati, "Can Hybrid Homomorphic Encryption Schemes be Practical?", *Proceedings of 5th International Conference on Multimedia Computing and Systems*, pp.**1-7, 2016.**

[18] Y.S. Gunjal, M.S. Gunjal and A.R. Tambe, "Hybrid Attribute Based Encryption and Customizable Authorization in Cloud Computing", *Proceedings of International Conference On Advances in Communication and Computing Technology*, pp.**1-5, 2018.**

[19] K. Raja and S. Pushpa, "Novelty-Driven Recommendation by using Integrated Matrix factorization and Temporal Aware Clustering Optimization", *International Journal of Communication Systems*, pp.**1-16, 2018.**

[20] N. Lee, Z. Chen and F. Chen, "Cloud Server Aided Computation for ElGamal Elliptic Curve Cryptosystem", *Proceedings of IEEE 37th Annual Computer Software and Applications Conference Workshops*, pp.**22-26, 2013.**

[21] R. Nivedhaa and J. Justus, "A Secure Erasure Cloud Storage System using Advanced Encryption Standard Algorithm and Proxy Re-Encryption", *Proceedings of International Conference on Communication and Signal Processing*, pp.**1-6, 2018.**

[22] X. Song and Y. Wang, "Homomorphic Cloud Computing Scheme based on Hybrid Homomorphic Encryption", *Proceedings of International Conference on Computer and Communications*, pp.**13-16, 2017.**

[23] A. Sude and V. Shinde, "Authenticated CRF Based Improved Ranked Multi-Keyword Search for Multi-Owner Model in Cloud Computing", *Proceedings of International Conference on Computing, Communication, Control and Automation*, pp.**1-5, 2017.**

[24] M. Thangapandiyan, P.M. Anand and K.S. Sankaran, "Enhanced Cloud Security Implementation Using Modified ECC Algorithm", *Proceedings of International Conference on Communication and Signal Processing*, pp.**12-17, 2018.**

[25] D.R. Kumar Raja and S. Pushpa, "Diversifying Personalized Mobile Multimedia Application Recommendations through the Latent Dirichlet Allocation and Clustering Optimization", *Multimedia Tools and Applications*, pp.**1-20, 2019.**

[26] A. Yun, J.H. Cheon and Y. Kim, "On Homomorphic Signatures for Network Coding", *IEEE Transactions on Computers*, Vol.**59**, Issue.**9**, pp.**1295-1296, 2010.**

[27] Z. Erkin, A. Piva, S. Katzenbeisser R.L. Lagendijk, J. Shokrollahi, G. Neven, M. Barni, "Protection and retrieval of encrypted multimedia content: when cryptography meets signal processing", *EURASIP Journal on Information Security **2007** (2007)*, Article ID 78943, doi:10.1155/2007/78943.

AUTHORS PROFILE

Madhira Srinivas: He got his B.Tech. and M.Tech. from NIT, Warangal and JNTUH University, Hyderabad in 2000 and 2003 respectively. Now, he is a research scholar in Computer Science & Engineering Department of JNTUH University, Hyderabad. His research areas of interest are cryptography, image processing, cloud computing, networks, operating systems and network security. He is a life member of ISTE since 2013. He has published more than 20 papers in various national and international journals. He presented papers in international conferences. He has 21 years of teaching experience and 10 years of research experience.

P. Sammulal: He got his B.Tech from Osmania University. M.Tech and PhD from JNTUH University, Hyderabad in the year 2002 and 2009 respectively. He has 25 years of teaching experience and 15 years of research experience. He is a life member of ISTE & CSIR and member of IEEE. He published more than 100 research papers in reputed national and international journals. He also presented papers in several IEEE international conferences. He acted as a session chair in several national conferences. He is a chairman of various inspection committees formed by JNTUH University, Hyderabad. He is a Board of Studies (BoS) member for several engineering colleges in the state of Telangana. His research areas of interest are cloud computing, cloud security, artificial intelligence, cryptography, deep learning and machine learning. He has received several awards and felicitated by the state of Telangana and JNTUH University, Hyderabad.