Enhanced Privacy Preservation of Cloud Data by using ElGamal Elliptic Curve (EGEC) Homomorphic Encryption Scheme

M.vedaraj[1*], and P.Ezhumalai[2]
[1] Research scholar
R.M.D. Engineering College, India
[e-mail: vedaraj84@gmail.com]
[2] HOD / Professor
R.M.D. Engineering College, India
[e-mail: ezhumalai.es@gmail.com]
*Corresponding author: M.vedaraj

Received February 24, 2020; revised April 12, 2019; revised July 9, 2019; accepted August 5, 2020; published November 30, 2020

Abstract

Nowadays, cloud is the fastest emerging technology in the IT industry. We can store and retrieve data from the cloud. The most frequently occurring problems in the cloud are security and privacy preservation of data. For improving its security, secret information must be protected from various illegal accesses. Numerous traditional cryptography algorithms have been used to increase the privacy in preserving cloud data. Still, there are some problems in privacy protection because of its reduced security. Thus, this article proposes an ElGamal Elliptic Curve (EGEC) Homomorphic encryption scheme for safeguarding the confidentiality of data stored in a cloud. The Users who hold a data can encipher the input data using the proposed EGEC encryption scheme. The homomorphic operations are computed on encrypted data. Whenever user sends data access permission requests to the cloud data storage. The Cloud Service Provider (CSP) validates the user access policy and provides the encrypted data to the user. ElGamal Elliptic Curve (EGEC) decryption was used to generate an original input data. The proposed EGEC homomorphic encryption scheme can be tested using different performance metrics such as execution time, encryption time, decryption time, memory usage, encryption throughput, and decryption throughput. However, efficacy of the ElGamal Elliptic Curve (EGEC) Homomorphic Encryption approach is explained by the comparison study of conventional approaches.

Keywords: Homomorphic, Privacy Preservation, Message Encoding, Encryption, Decryption, Message Decoding.

http://doi.org/10.3837/tiis.2020.11.015 **ISSN : 1976-7277**

1. Introduction

Cloud computing is currently the over used buzzword in the software industry. Cloud provides many services to a customer over the network such as storage, application and database [1]. Cloud computing has different attractive characteristics such as availability, high reliability, high scalability, multi-sharing and agility. It has many advantages in terms of lower IT infrastructure cost, less maintenance cost, lower software cost and unlimited storage capacity [2].

Nowadays, cloud computing is one among the hot topics in both industry and academic research [3] like storage, security, load balancing, task assignment, etc. Secure data access and maintaining the confidentiality of data stored in a cloud are the most important research areas in cloud computing. Privacy preservation [4] is a process of safeguarding the secret information from unauthorized access which is stored in a cloud. Privacy of cloud data can be ensured by some security methods like generating keys, encryption of data and decryption of data[5]. Many privacy preservation mechanisms are developed to guarantee the confidentiality of data like encryption based methods, Access control based mechanisms, query integrity/keyword searches schemes, auditability schemes, etc. But there are some difficulties like multiple user access, cost-efficiency, inadequate big data storage and parallel computing techniques which does not support the integration of big data with cloud computing [6,7].

To overcome the privacy issues in cloud storage, homomorphic encryption scheme is proposed.Unlike conventional encryption algorithms like RSA or AES, Homomorphic Encryption(HE) is a type of symmetric key encryption which allows to perform specific types of operations like addition or multiplication on enciphered data. Homomorphic Encryption (HE) is a most popular powerful tool that helps to preserve the data privacy in many applications such as internet voting, encrypted query processing, daily data usage, health care system etc. Homomorphic Encryption solves confidentiality problems and problems related to personal records like medical records. HE reduces computational overheads and ensures the confidentiality of information stored in cloud storage.

In the cloud framework, the multi-cloud model can be used in which private cloud is used to store personal information and public cloud is used to store encrypted data. Here, CSP is a mediator between user of the data and cloud storage [6]. The proposed scheme ElGamal Elliptic Curve(EGEC) encryption algorithm for storing data into the cloud in a secure way. This technique allows the user to do various homomorphic operations such as addition/ multiplication on encrypted point. The essential contributions of proposed methodology are:

1.Access policy verification is performed to limit data access.
2.The input data is encrypted using the ElGamal Elliptic Curve (EGEC) homomorphic encryption algorithms which improves data privacy.
3.Then, homomorphic operations are computed on the enciphered data.
4.ElGamal Elliptic Curve (EGEC) decryption and homomorphic operation are performed to get back the original plaintext.

The resting research article is organized as follows: In section 1, various encryption/decryption mechanisms are reviewed and their merits and demerits are discussed. In section 2, flow diagram for the proposed research work is briefly explained. In section 3,

experimental results of proposed mechanisms are illustrated. Suitable algorithms for the mechanisms are discussed in section 4. Finally, conclusion are described in section 5.

2. Related Work

In this section, various encryption and decryption mechanisms are surveyed with their major advantages and disadvantages.

Hayward and Chiang[8] developed an algorithm called parallel processing of Gentry Fully Homomorphic encryption. In this paper, issues in privacy preservation of cloud data is solved and the speed of data encryption is increased. Dhote[9] proposed a algorithm named as Fully Homomorphic Encryption (FHE) which addresses the different security problems such as data security, integrity, privacy, confidentiality, and authentication. The proposed FHE scheme reduces the size of the encrypted data which makes data processing efficient. Cao et al[10] presented a solution for solving the problems of multi-keyword rank search over encryption and also suggested various privacy requirements to safeguard the data stored in cloud. In this article, a coordinate matching scheme was selected among the multi-keyword concepts to determine the similarities among various threat models. The motive of this proposed methodology is to protect the confidential information which is being outsourced to the data user. Important advantages of this scheme is both communication and computation time is low.

Kaaniche and Laurent[11] surveyed issues related to security, privacy and various cryptographic techniques used in cloud storage systems. The important motive of this reseach survey is to effectively compare the performance metrics of different cryptographic techniques used in the cloud storage system. Yu et al[12] suggested an ID- based RDIC protocol used to safeguard the confidentiality of information which is stored in a cloud data storage system . This research work achieves the correctness of the protocol and no information leakage in a file that is stored in cloud storage. The cost of ID-based remote data integrity checking is examined in the matter of computation, communication, and storage. Tang et al[13] surveyed various security threats such as loss of data, unauthorized access to data, corruption of data, etc. He also analyzed various security requirements like data confidentiality, data access controllability, data integrity, etc. At the end, authors provide the corresponding solution to the cloud data services for resolving the issues of confidentiality guarantee, controlling data access and privacy of data stored in a cloud.

Dasgupta et al[14] proposed an algorithm named as Homomorphic Encryption (HE) based on the polynomial ring for performing operations on encipher plaintext. The proposed algorithm uses homomorphic encryption only to some extent. To makes this algorithm a fully homomorphic, refresh mechanism is used. El Makkaou et al[15] suggested an algorithm named FS-RSA which is used to raise the speed of encryption and decryption process. FC–RSA algorithm consists of four different components such as generation of keys, encryption of plaintext, decryption of ciphertext and evaluation of data.The major advantages of this work are greater performance and reduced execution time. Farokhi et al[16] presented an algorithm called semi homomorphic encryption which allows performing some operations on enciphered

information. In this research article, pallier encryption algorithm is implemented to ensure the stability and certain bounds on closed loop performance. Zhang et al [17] suggested a solution for the local recording problem. In this paper, the appropriate method of clustering is been selected for solving the Proximity-Aware Clustering Problem of Local-Recoding Anonymization. Also, the algorithms are designed with map-reduce to perform parallel computation.

Yu et al[18] presented a mechanism named multiple keywords similar to other searches which ensures the privacy of cloud data. In this mechanism, text data is considered as an input where the user specifies the multiple keywords and cloud returns a file that contains keywords. User access privacy is ensured by using a blind signature protocol. Bloom filter's bit pattern technique is used to increase the speed of searching at the cloud and to ensure a secure search against insider attack. Wang et al[19] proposed a methodology called privacy-preserving multi keyword fuzzy search which make use of locality-sensitive hashing approach to guarantee privacy to the cloud data. The proposed research work removes the need for predefined dictionary and the primary objective is to protect user data privacy which guarantees privacy of content in a file, privacy of index and privacy of user query.

Kai Fan et al[20] proposed more efficient access control system based on privilege protection. Two different and efficient mechanisms implements read and write access control in both private and public domains such as key aggregate based encryption(KABE) and hierarchy attribute based encryption(HABE). The considerable amount of access efficiency is improved by key aggregate based encryption(KABE) scheme and also safeguards the data from unautorized access using ABE scheme.

3. Proposed system

In this research article, a new efficient mechanism ElGamal Elliptic Curve (EGEC) is proposed for safeguarding the secresy of user data stored in cloud storage system. In this environment, consider a multi-cloud model shown in **Fig. 1**. It incorporate the proficiency of both private cloud and public cloud. The encrypted form of user information is stored in private cloud by using EGEC approach and data stored in a public cloud is shared by the cloud storage which is accessed on the limited access using access policy. Actually, homomorphic scheme is a symmetric cryptography algorithm. However, in the proposed mechanism, we implement homomorphic scheme as an asymmetric cryptography algorithm as EGEC is converted with respect to the capabilities of homomorphic scheme and it uses key size of 512 bits. In this technique, for secure data storage and retrieval, a multi-cloud environment is used which addresses various security issues like data integrity, service availability, etc. The proposed mechanism consists of

- Access policy verification
- Key generation
- Message encoding
- Encryption
- Evaluation
- Decryption
- Message decoding

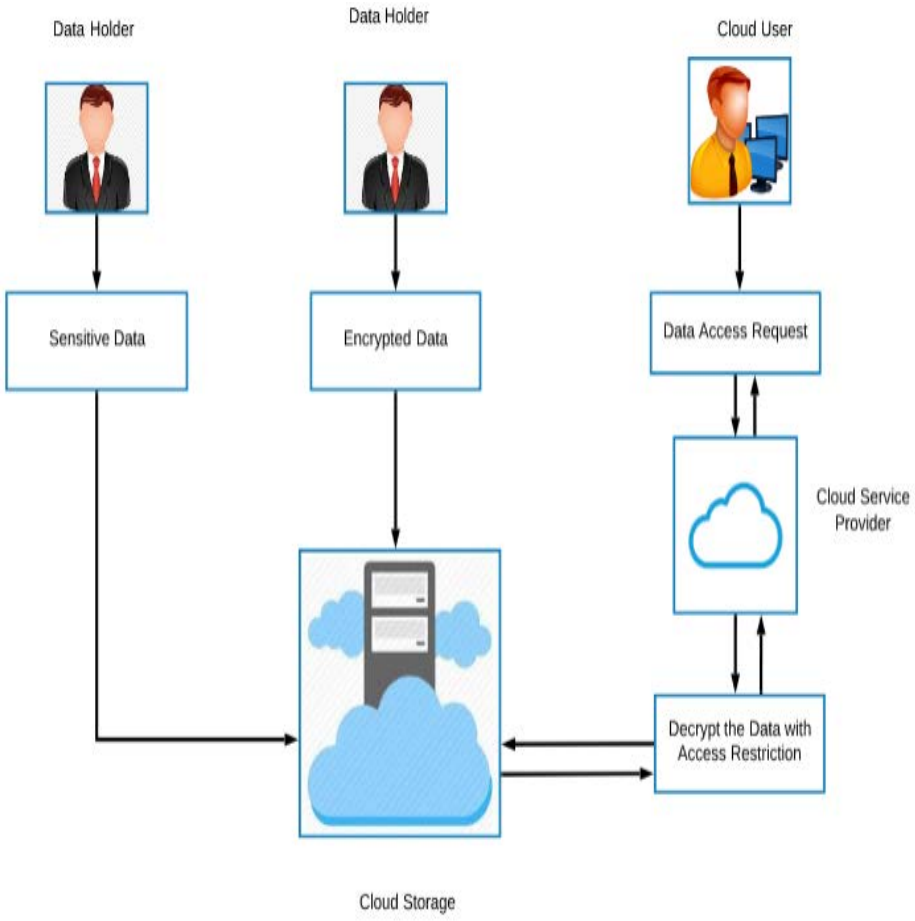

Fig. 1. Multi Cloud Model Architecture

The Flow diagram for the proposed scheme is shown in **Fig. 2**. User wants to access data stored in a cloud storage, user passes a request to cloud storage. The cloud server verifies the access policy to limit the access to cloud data. After restricting the access, then keys are generated. In this work, message encoding is used for the representation of messages into Elliptic curve points. The ElGamal Elliptic Curve (EGEC) Homomorphic encryption algorithm
is used to encrypt these points. Then, fully homomorphic operations are applied to the encrypted points and they are forwarded to the requested user. Atlast, homomorphic operations and ElGamal Elliptic Curve (EGEC) decryption algorithm is used to generate back an original point. Finally, Message decoding is performed to represent the Elliptic curve points into a message. Thus, this proposed work uses multi-cloud environment to guarantee a confidentiality of cloud information.

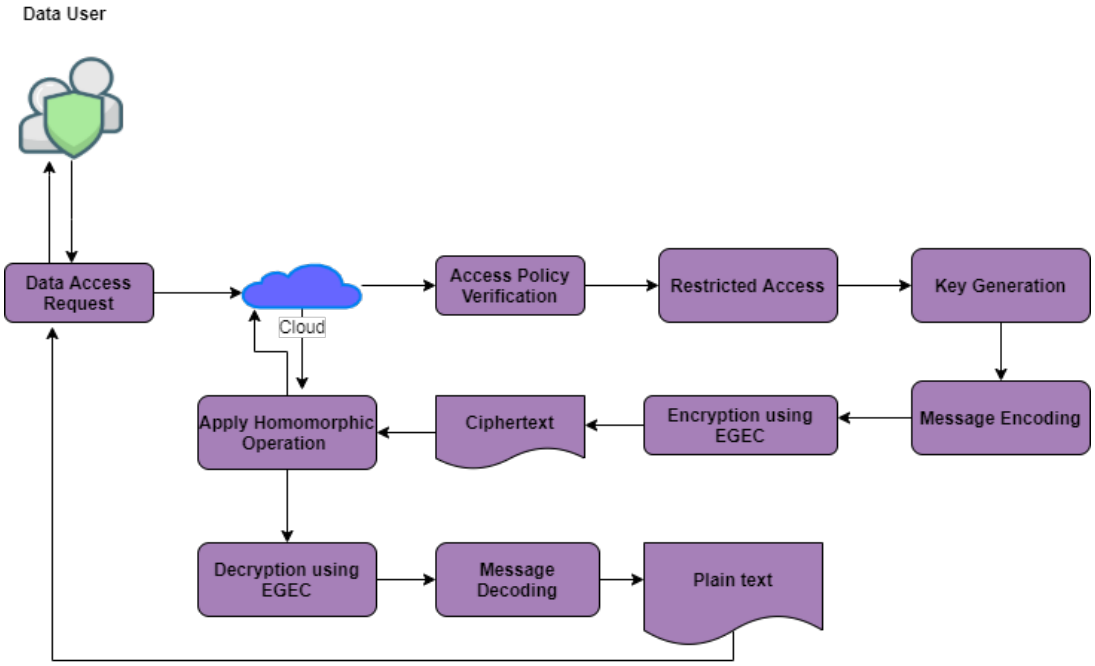

Fig. 2. Architecture of proposed system

3.1 Access Control Policy Generation

Consider a hospital, hospital administrator is the data owner, who has the access to control the complete security system and assign roles to the users and authorizes a unique password for every user. The cloud service provider (CSP) acts as a link between the data user and hospital administrator. In Hospital environment, data users are Receptionists, Doctors, Nurse and Patients. Whenever a data user passes a request to access data in cloud storage, the cloud server validates the access policy of the request and then the access permission is granted to user to get the requested data from cloud. Different roles and privileges of a data user in a hospital are shown in **Table 1**. Add Patient, Read / Write Prescription, Read / Write Patient Record is some of the privileges of data user. The receptionist has the privilege to add a new patient record and modify the existing patient details. However, the receptionist cannot have access to the patient records. The doctor has the privilege to Read/Write Patient medical records and Read/Write prescriptions. The Nurse has the privilege to read a prescription. Finally, the patients have the privilege to read his/her own medical records.

Table 1. Access Control policy generation

Roles	Receptionist	Doctor	Nurse	Patient
Patient Registration	Read&Write	No Access	No Access	No Access
Patient Medical Record	No Access	Read&Write	Read	Read
Patient Prescription	No Access	Read&Write	Read	Read

3.2 Key Generation

In Cryptography, Key generation is an algorithm which is used for generating keys. Generated keys are mainly used for data encryption and decryption of data. The private key (key_{pr}) is computed by choosing 'k' randomly from the range [1, n-1]. The public key (key_{pu}) is computed using key_{pr} and Generator function G_c .

Algorithm I: Key Generation Algorithm

Input: Random Number (k) and Generator function (G_c)

Output: Private key (key_{pr}) and public Key (key_{pu})

Step1: Private key is computed by selecting a random number k from [1, n-1].

Private key (key_{pr}) =k

Step2: Compute public key (key_{pu}) = key_{pr}*G_c . where G_c is a Generator function.

3.3 Message Encoding

Message Encoding is defined as the process of transforming a message into elliptic curve points. ElGamal Elliptic Curve (EGEC) schemes can encrypt and decrypt only points on the elliptic curve, not a input text data. In proposed methodology, consider an Input text message to be encrypted. The input message is divided into a block of fixed size where each block size is of one character. Then, each character in a text is converted into ASCII value and each ASCII value is directly mapped into Elliptic Curve points.

Algorithm II: Message Encoding Algorithm

Input: PlainText message

Output: Elliptic Curve points

Step 1: Divide the message into block of fixed size where the block size is one character.

Step 2: Convert each character in a text message to its corresponding ASCII value.

Step 3: ASCII values are directly mapped into points on Elliptic Curve.

3.4 Encryption

Encryption is a method by which plain text is converted into cipher text. In this proposed work, Elgamal – Elliptic curve (EGEC) Homomorphic encryption scheme can encrypt the plain text points by increasing security with minimal cost. In this algorithm, private key (key_{pr}), public key (key_{pu}) and point on curve (P_m) for a particular block of message are given as input. Then, cipher point 'C_p' is generated using 4-bit random number 'r_n' and generator function 'G_c' . After that, the point on curve 'P_m' for a particular block of message is encrypted by using

random number 'r_n' and public key (key_{pu}). Homomorphic computations are carried out on a set of encrypted points 'E_p' by using either addivtive property or multiplicative property. Finally, the encrypted point (E_p) is stored into cloud.

Algorithm III –ElGamal Elliptic Curve (EGEC) Homomorphic Encryption

Input: Private key (key_{pr}), Public key (key_{pu}), Generator function (Gc)
Output: Encrypted point (E_p)
Procedure:
Step 1: Encrypted point E_p is computed as
$E_p \leftarrow (r_n * key_{pu}) + P_m$
where P_m is a point on curve for a particular block of message.
Step 2: Homomorphic computation is performed on set of encrypted points E_p
$E_p = (r_n * key_{pu}) + \sum P_m$
Where n represents number of block of messages.
Step 3: Finally, the encrypted point (E_p) is uploaded to the cloud.

3.5 Decryption

Decryption is the method by which cipher text can be converted into a plain text. In this proposed work, homomorphic computation like addition or multiplication operations are carried out to extract the cipher point from the encrypted point. For decryption, Elgamal Elliptic Curve decryption algorithm is used in which private key (key_{pr}) and encrypted point (E_p) are given as input to the algorithm. Using a private key (key_{pr}), the cipher point 'C_p' and random value'r_n' can be extracted encrypted point(E_p) and 'G_c' respectively. Then, the points are decrypted .

Algorithm IV –ElGamal Elliptic Curve (EGEC) Homomorphic decryption

Input: Private key (key_{pr}), Encrypted point (E_p)
Output: Decrypted point (D_p)
Procedure:

Step 1: Consider a private key (key_{pr}) for decryption.
Step 2: Consider the encrypted point (E_p) and extract the cipher point C_p .
$C_p \leftarrow key_{pr} * E_p$
Step 3: Extract the random value r_n from the generator function G_c.
$C_p \leftarrow key_{pr} * (r_n * G_c)$
$C_p \leftarrow r_n * (key_{pr} * G_c)$
$C_p \leftarrow r_n * key_{pu}$
Step 4: Compute decrypted point as $D_P = E_p – key_{pr} * C_p$

3.6 Message Decoding

Message decoding is a method for transforming Elliptic Curve Points into the message. To convert encrypted Elliptic curve points into message, let us consider an decypted point 'Pt_m' and set 'm' to be the largest integer and less than x. Then, each decrypted point (x_p,y_p) is

decoded as the symbol ‘m’ and is converted into a message.

Algorithm V - Message Decoding Algorithm

Input: Elliptic Curve points
Output: PlainText Message
Step 1: Let us Consider a decrypted point $Pt_m = (x_p, y_p)$.
Step 2: Set m to be the largest integer but less than x. Then, point (x_p, y_p) decodes as the symbol m.
Step 3: Transform m to plaintext message

4. Experimental results and discussion

In this proposed research work, Open Stack cloud environment can be used to analysis the security measures discussed in Section 4.1. The OpenStack is a popular open source cloud operating systems that supports different cloud environments. The host machine corresponds to the following characteristics: minimum 2GB of RAM, minimum 20 GB disk space and access to the Internet and Processors with hardware virtualization extensions. The performance analysis of the proposed ElGamal Elliptic Curve (EGEC) Homomorphic encryption scheme and existing schemes are discussed in section 4.1

4.1 Performance analysis

The experimental outcomes for both existing and proposed schemes are tested using numerous performance metrics such as execution time, encryption and decryption time, memory usage, and encryption and decryption throughput. The existing schemes used in this research work are matrix operation for randomization and encryption (MORE) and polynomial operation for randomization and encryption (PORE).

4.1.1 Execution time

Execution time is defined as the amount of time taken by the algorithm for generating cipher text from plain text and vice versa in cloud storage. The excecution time comparison between existing schemes and proposed EGEC scheme based on various key sizes are shown in **Fig. 3**. From the execution time analysis in **Table 2**, it is clear that the existing approach MORE has taken 144ms, 151ms, 179ms, 205ms, 230ms and 270ms to complete encryption and decryption for the key sizes 32bits, 64 bits, 128bits, 256bits, 512bits, 1024bits respectively and PORE has
taken 139ms, 145 ms,170 ms, 200ms, 240ms, and 280ms to complete encryption and decryption for the key sizes 32bits, 64bits, 128bits, 256bits, 512bits and 1024bits respectively. The proposed ElGamal Elliptic Curve (EGEC) Homomorphic approach is found to have the lowest execution time as it takes nearly 100ms, 140ms, 162ms, 196ms, 200ms and 263ms to complete encryption and decryption.

Table 2. Comparison of Execution time

Key Size(Bits)	Execution Time(ms)		
	MORE	PORE	EGEC
32	144	139	100
64	151	145	140
128	179	170	162
256	205	200	196
512	230	240	200
1024	270	280	263

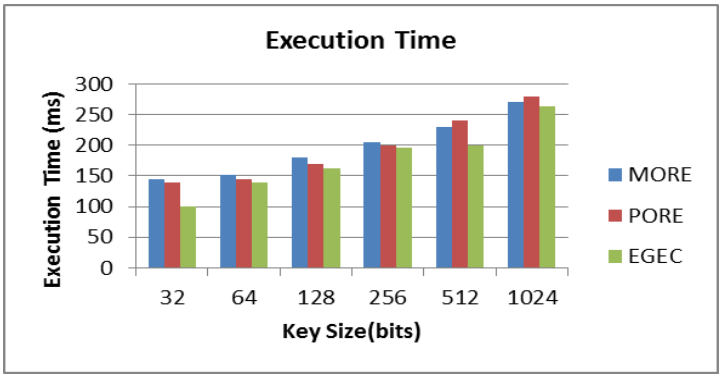

Fig. 3. Comparison of Execution time

4.1.2 Encryption Time

Encryption time is described as the amount of time taken by the algorithm to convert the input text into cipher text. The encryption time can be expressed in terms of milliseconds. The encryption time comparison between existing schems and proposed EGEC scheme are shown in Figure-4. From **Table 3**, existing schems MORE takes 238ms, 350ms, 420ms, 524ms and 625ms for encrypting the input size of 512KB, 1024KB, 1536KB, 2048KB and 2560KB. and PORE takes 224ms, 325ms, 400ms, 512ms and 621ms for encrypting the input size of 512KB, 1024KB, 1536KB, 2048KB and 2560KB. The proposed EGEC homomorphic scheme takes 198ms, 320ms, 380ms, 500ms and 602ms for encrypting the input size of 512KB, 1024KB, 1536KB, 2048KB and 2560KB. The proposed ECEC homomorphic scheme has reduced encryption time when compared to existing schemes.

Table 3. Comparison of Encryption and Decryption time

Input Size(KB)	Encryption Time(ms)			Decryption Time(ms)		
	MORE	PORE	EGEC	MORE	PORE	EGEC
512	238	224	198	220	210	180
1024	350	325	320	280	279	250
1536	420	400	380	385	378	350
2048	524	512	500	498	470	450
2560	625	621	602	590	575	495

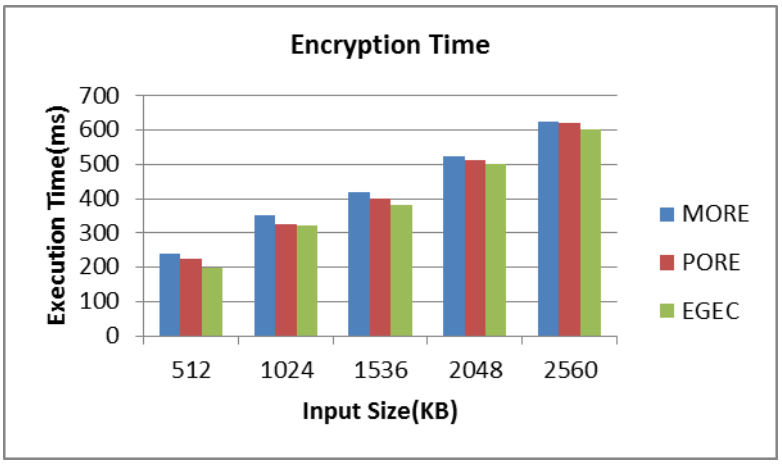

Fig. 4. Comparison of Encryption time

4.1.3 Decryption time

Decryption time is the time taken by the algorithm to convert the cipher text into plain text. The decryption time can also be expressed in terms of milliseconds. The decryption time comparison between existing schemes and proposed EGEC scheme are shown in Figure-5. From **Table 3**, existing schems MORE takes 220ms, 280ms, 385ms, 498ms and 590ms for decrypting the input size of 512KB, 1024KB, 1536KB, 2048KB and 2560KB and PORE takes 210ms, 279ms, 378ms, 470ms, 575ms for decrypting the input size of 512KB, 1024KB, 1536KB, 2048KB and 2560KB. The proposed EGEC homomorphic scheme takes 180ms, 250ms, 350ms, 450ms, 495ms for decrypting the input size of 512KB, 1024KB, 1536KB, 2048KB and 2560KB. The proposed EGEC homomorphic scheme has reduced decryption time when compared to existing schemes.

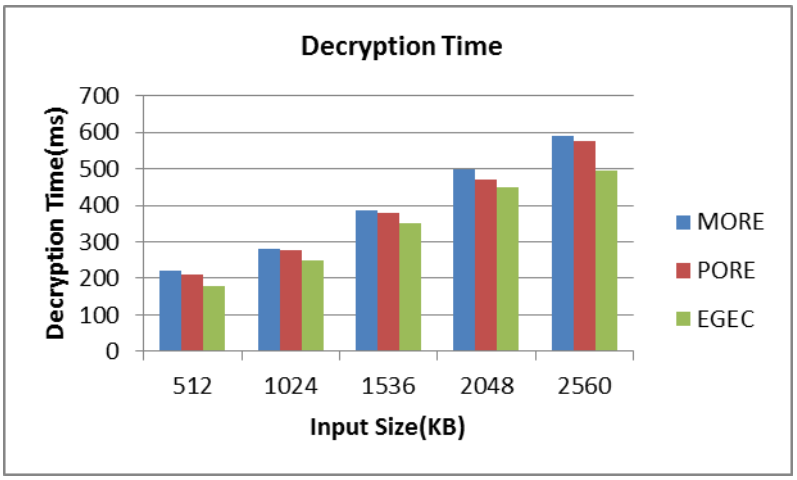

Fig. 5. Comparison of Decryption time

4.1.4 Memory usage

Memory usage is defined as the amount of memory space required to implement the encryption and decryption algorithms. The existing scheme MORE consumes 43.2 KB of memory and PORE consumes 32.8 KB of memory. The proposed EGEC homomorphic scheme taken only 26.2 KB of memory. Figure-6 shows the memory usage of existing approaches MORE, PORE and proposed EGEC homomorphic approach. From the memory usage analysis, our proposed approach EGEC Homomorphic algorithm taken less memory.

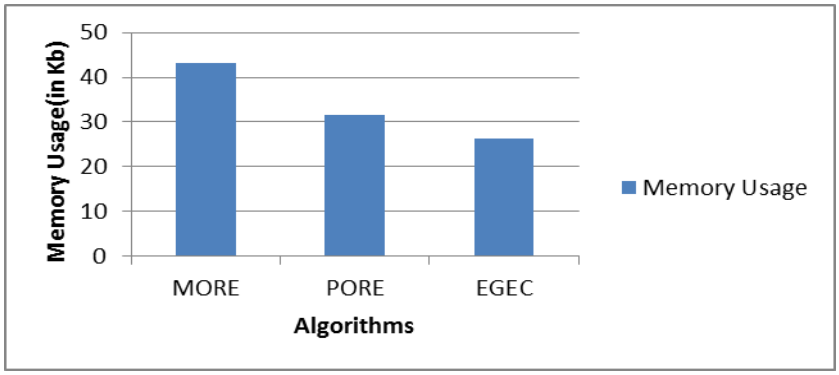

Fig. 6. Comparison of Memory Usage

4.1.5 Encryption Throughput

Encryption throughput can be defined as the process of dividing the size of the plain text using encryption time. Encyption Throughput is used to compute the speed of encryption process. **Fig. 7** shows encryption throughput of existing approaches MORE, PORE and proposed EGEC approach. From the encryption throughput result analysis, it is clear that the proposed ElGamal Elliptic curve (EGEC) homomorphic encryption algorithm has the highest throughput than other existing schemes.

Table 3. Comparison of Encryption and Decryption Throughput

Input Size(KB)	Encryption Throughput(ms)			Decryption Throughput(ms)		
	MORE	PORE	EGEC	MORE	PORE	EGEC
512	2.15	2.28	2.58	2.32	2.43	2.84
1024	2.92	3.15	3.20	3.65	3.67	4.09
1536	3.65	3.84	4.04	3.98	4.06	4.38
2048	3.90	4.00	4.09	4.11	4.35	4.55
2560	4.09	4.12	4.25	4.33	4.45	5.17

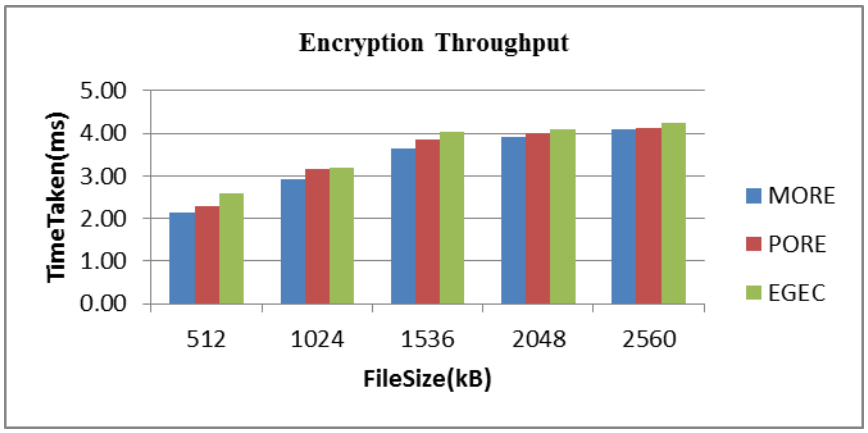

Fig. 7. Comparison of Encryption Throughput

4.1.6 Decryption Throughput

Decryption throughput can be defined as the process of dividing the size of the encipher text using decryption time. Decyption throughput used to compute the speed of decryption process. The decryption throughput comparison between existing schemes and proposed EGEC scheme are shown in **Fig. 8**. From the decryption throughput analysis, it is clear that the proposed approach ElGamal Elliptic Curve (EGEC) homomorphic decryption algorithm has the highest throughput than other existing schemes.

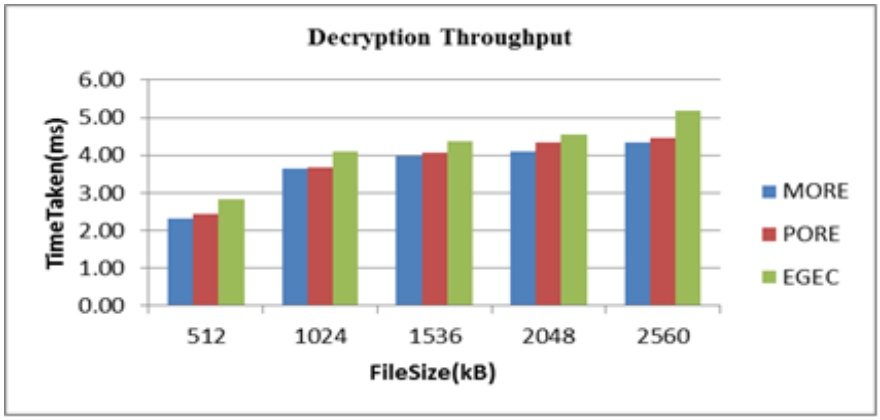

Fig. 8. Comparison of Decryption Throughput

5. Conclusion

In this research paper, to guarantee the privacy of cloud data, we proposed a new scheme named ElGamal Elliptic Curve (EGEC) encryption algorithm. This proposed approach consists of access policy verification, key generation, public key verification, message encoding, encryption, decryption and message decoding. Also, it has the entities like Receptionist, Doctor, Nurse, and Patient. The data user wants to access data from cloud and sends a data access permission request to cloud storage. user access rights is validated by the CSP and then grant permision to access data. Message encoding is used to convert the

message into Elliptic curve points which are encrypted using the EGEC Homomorphic algorithm. Homomorphic operations such as addition opearation and multiplication operation are performed on encrypted points. Decryption and message decoding are performed to get back the original message. The important advantage of the EGEC technique is guarantees the confidentiality of the secrete data. The proposed mechanism is also compared with existing mechanisms like MORE, PORE. Different performance measures like execution time, encryption time, decryption time, memory usage, encryption throughput and decryption throughput are used to examine the proposed approach. Thus the proposed scheme ensures both the security and privacy of cloud data. From the experimental result anlysis, it is observed that the proposed EGEC scheme efficiently reduces the complication of encryption and decryption.

References

[1] Kadam Prasad, Jadhav Poonam, Khupase Gauri, N.C.Thoutan, "Data sharing security and privacy preservation in cloud computing," in *Proc. of Green Computing and Internet of Things(ICGCIoT), 2015 International Conference*, pp. 1070–1075, 2015. Article (CrossRef Link)

[2] Zhang .X, Chang Liu, Surya Nepal, Chi Yang, and Jinjun Chen, "Privacy preservation over big data in cloud systems," in *Proc. of Security, Privacy and Trust in Cloud Systems, Springer*, pp. 239–257, 2014. Article (CrossRef Link)

[3] J. Suganthi, J. Ananthi J, S. Archana, "Privacy preservation and public auditing for cloud data using ASS in multi-cloud," in *Proc. of Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference*, pp. 1–6, 2015. Article (CrossRef Link)

[4] Wei Wang, Lei chen, and Qian Zhang, "Outsourcing high-dimensional healthcare data to cloud with personalized privacy preservation," *Computer Networks*, Vol.88, pp.136–148, 2015. Article (CrossRef Link)

[5] Frank Li, Richard Shin, and Vern Paxson, "Exploring privacy preservation in outsourced k-nearest neighbors with multiple data owners," in *Proc. of the 2015 ACM Workshop on Cloud Computing Security Workshop*, pp. 53–64, 2015. Article (CrossRef Link)

[6] Hong Liu, Huansheng Ning , Qingxu Xiong , and Laurence T. Yang, "Shared authority based privacy-preserving authentication protocol in cloud computing," *IEEE Trans. Parallel Distrib. Syst*, Vol.26, No.1, pp.241–251, 2015. Article (CrossRef Link)

[7] Qinghua Shen, Xiaohui Liang, Xuemin Shen, Xiaodong Lin, and Henry Y. Luo, "Exploiting geo-distributed clouds for a e-health monitoring system with minimum service delay and privacy preservation," *IEEE J. Biomed. Health Inform*, Vol.18, No.2, pp. 430–439, 2014. Article (CrossRef Link)

[8] Ryan Hayward and Chia-Chu Chiang, "Parallelizing fully homomorphic encryption for a cloud environment," *J. Appl. Res. Technol*, Vol. 13, No.2, pp.245–252, 2015. Article (CrossRef Link)

[9] Manish M.Potey, A.Dhote, Deepak H, and Sharma, "Homomorphic encryption for security of cloud data," *Procedia Comput. Sci*, Vol.79, pp.175–181, 2016. Article (CrossRef Link)

[10] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst*, Vol.25, pp.222–233, 2014. Article (CrossRef Link)

[11] Kaaniche. N, Laurent. M, "Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms," *Comput. Commun.*, Vol. 111, pp.120–141, 2017. Article (CrossRef Link)

[12] Yong Yu, Man Ho Au, Giuseppe Ateniese, Xinyi Huang, Willy Susilo, Yuanshun Dai, and Geyong Min, " Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," *IEEE Trans.Inf. Forensics Secur*, Vol. 12, pp.767–778, 2017. Article (CrossRef Link)

[13] Jun Tang, Yong Cui, Qi Li, Kui Ren, Jiangchuan Liu, and Rajkumar Buyya "Ensuring security and privacy preservation for cloud data services," *ACM Comput. Surv. (CSUR)*, Vol. 49, No.1, pp.13, 2016. Article (CrossRef Link)

[14] Smaranika Dasgupta, and S.K.Pa, "Design of a polynomial ring based symmetric homomorphic encryption scheme," *Perspect. Sci*, Vol. 8, pp.692–695, 2016. Article (CrossRef Link)

[15] Khalid El, Makkaoui Abderrahim, Beni-Hssane, Abdellah Ezzati, Anas El-Ansari, "Fast cloud-RSA scheme for promoting data confidentiality in the cloud computing," *Procedia Comput.Sci.*, Vol.113, pp.33–40, 2017. Article (CrossRef Link)

[16] Farhad Farokhi, Iman Shames, and Nathan Batterham, "Secure and private cloud-based control using semi-homomorphic encryption," *IFAC-Papers Online*, Vol. 49, No.22, pp.163–168, 2016. Article (CrossRef Link)

[17] Xuyun Zhang, Wanchun Dou, Jian Pei, Surya Nepal, Chi Yang, Chang Liu, and Jinjun Chen, "Proximity-aware local-recoding anonymization with map reduce for scalable big data privacy preservation in cloud," *IEEE Trans. Comput*, Vol. 64, pp. 2293–2307, 2015. Article (CrossRef Link)

[18] Chia-Mu Yu ,Chi-Yuan Chen , and Han-Chieh Chao, "Privacy-Preserving Multikeyword Similarity Search Over Outsourced Cloud Data," *IEEE Systems Journal*, Vol.11, No.2, pp.385-394, 2017. Article (CrossRef Link)

[19] Bing Wang, Shucheng Yu, Wenjing Lou, and Y. Thomas Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Proc. of INFOCOM, 2014 Proceedings IEEE*, pp. 2112–2120, 2014.Article (CrossRef Link)

[20] Kai Fan, Qiong Tian, Junxiong Wang, Hui Li, Yintang Yang, "Privacy Protection Based Access Control Scheme in Cloud-Based Services," *IEEE 2017*, Vol.14, No. 1, pp.61-71, 2017.

[21] Khalil Hariss, Hassan Noura, and Abed Ellatif Samhat, "Fully enhanced homomorphic encryption algorithm of MORE approach for real world applications," *J. Inf.Secur. Appl*, Vol.34, No.2, pp. 233–242, 2017. Article (CrossRef Link)

Mr.VEDARAJ, B.Tech, M.E is Assistant Professor in Department of Computer Science and Engineering at R.M.D. Engineering College. He obtained his B.E(CSE) from St.Peters Engineering College and M.E(CSE) from DMI College of Engineering. His research interests includes cloud computing and IOT .

Dr.P.Ezhumalai B.E, M.Tech, Ph.D is the Professor and Head of the Department of Computer Science and Engineering at R.M.D Engineering College, Chennai. He completed his B.E (CSE) in the year 1992 from University of Madras and M.Tech (CSE) in the year 2006 from J.N.T. University Hyderabad and Ph.D in Design and Implementation of High Performance Architecture for NoC System from Anna University, Chennai in the year 2012. He has 25 Years of working experience in the teaching profession and 10 years of research experience. He has National Citizenship Gold Medal Award for excellence from Global Economic Progress and Research Association, New Delhi.