Classical Approach Is To Perform Computation Using Trusted Third Party

Kanaparthi Pavani[1] Dr. Kondapalli Venkata Ramana[2]
[1]Student [2]Associate professor
[1,2]Department of Computer Science and System Engineering (CS&SE)
[1,2]Andhra University College of Engineering, Visakhapatnam, India

Abstract— In today's world of advanced technology, sharing private or confidential data over a network is not secure. Cryptography is a technique in information security by which we can protect our private data from unknown persons. In this paper, we focus on ellipstic based curve cryptography-based secure multi-party computation methods. The classic approach to SMC is to use a trusted third party to perform calculations. But when we talk about the actual situation, TTP is difficult to achieve, and it is imperative to eliminate TTP in SMC. Furthermore, existing solutions proposed for SMC use classical full-state encryption schemes such as RSA and Paillier. But such cryptosystems are expensive, so the resulting SMC protocol is not scalable. We propose an ECC-based SMC homomorphic encryption scheme based on performance metrics such as computational cost and communication cost. Among several selected algorithms, we recommend an efficient algorithm that provides security and reduces cost, and can be applied to many applications requiring privacy.

Keywords: Elliptic Curve Cryptography (ECC), Trusted Third Party (TTP), Cryptosystem

I. INTRODUCTION

Now we live in a digital world where our messages or information are transferred or exchanged instantaneously between different users or systems through communication channels. To make the most of this data requires collaborative computing on the data. But federated computing of data can pose a threat to the privacy of personal data. Therefore, some protocol is needed to overcome this problem. SMC solves this problem. There are many real-world situations where privacy can be an issue, some of which are worth mentioning. Consider an organization that has multiple cameras installed to capture moments of people in the area. The presence of a central administrator and the desire to perform joint computations on the recordings meant that no one passed through each camera. Therefore, in order to perform joint computations on data or records, each record is sent to the administrator in encrypted form. After getting the encrypted data, add it up and send the result to all parties. In general, privacy is the maintainer.

Existing homomorphic encryption based approaches uses classical public key cryptosystem. Be that as it may, in this methodology here does not require presence of private channel and furthermore guarantees abnormal state of security. Consider a typical medical research application where multiple hospitals want to conduct joint research on their patient data, but at the same time they must comply with government privacy policies and laws to prevent these hospitals from pooling their data due to the confidentiality of patient records. In this case, it is necessary to find a solution that enables hospitals to compute the required functions on the federation of their databases, without revealing their patient data.

Consider the interaction between different intelligence agencies; they do not allow each other to freely access data due to classified information; if they do, one mole in one agency has access to a large amount of resources. It can be harmful to institutions and can compromise safety. Thus, here we propose a protocol that eliminates TTP and minimizes the overhead associated with cryptography-based privacy-preserving methods.

The multiparty calculation issue was presented by Yao and reached out by Goldreich, Micali and Wigderson. They utilize the fundamental technique to speak to the issue as combinatorial circuit. Taking an interest parties at that point run a convention for each entryway in the circuit; subsequently it isn't adaptable for vast information sources. Goldreich presents different strategies to perform multiparty calculation.

For example, homomorphic encryption and mystery sharing. In writing different application, For example, security safeguarding information mining, private factual data recovery, privacy protecting information base access have been recommended that request SMC among gatherings.

II. BACKGROUND STUDY

The multiparty computation problem was introduced by Yao and extended by Goldreich, Micali and Wigderson. They use the basic method to represent the problem as combinatorial circuit. Participating parties then run a protocol for every gate in the circuit; hence it is not scalable for large inputs. Goldreich presents various other methods to perform multiparty computation such as homomorphic encryption and secret sharing .In literature various application such as privacy preserving data mining, private statistical information retrieval, Privacy preserving data base access have been proposed that demand SMC among parties .

A. Homomorphic Encryption:

Homomorphic encryption schemes allow parties to perform simple computation on encrypted data, as the computation are performed on encrypted data,the data is not revealed in the whole process and the privacy get maintained .Typically, a third party can calculate one of the encrypted sum or the encrypted product of two messages.

1) Properties of Homomorphic Encryption:

It has mainly two properties

a) Additive Homomorphic Encryption: A homomorphic encryption is additive, if
$E_k(PT1 \oplus PT2) = E_k(PK1) \oplus E_k(PK2)$

b) Multiplicative Homomorphic Encryption:

A homomorphic encryption is multiplicative, if

$$E_k(PK1 \otimes PK2) = E_k(PK1) \otimes E_k(PK2)$$

We refer additive homomorphic encryption scheme based on ECC in this paper.

B. Elliptic Curve Cryptography (ECC):

It is public key cryptography approach based on the algebraic structure of elliptic curves over finite fields .there are two types of finite fields where the elliptic curve is defined :binary field and prime fields. Prime fields Fp where p is a large prime number, and binary fields F2 m.

III. METHODOLOGY

Cryptography refers to secure information and communication techniques derived from mathematical concepts and a set of rule-based calculations called algorithms, to transform messages in ways that are hard to decipher. These deterministic algorithms are used for cryptographic key generation. Cryptosystems use a set of procedures known as cryptographic algorithms, to encrypt and decrypt messages to secure communications among computer systems, devices and applications. A cipher suite uses one algorithm for encryption, another algorithm for message authentication and another for key exchange. This process, embedded in protocols and written in software that runs on operating systems and networked computer systems, involves:

A. Public and Private key generation for data encryption/decryption.

Public key is used to encrypt the plain text to convert it into cipher text and Private key (secret key) is used by the receiver to decrypt the cipher text to read the message.

Implementation of project done by python django framework with ECC implementation. Considering two servers as Level -1 and Level- 2.

Level -1 : Secure Multi party Server used to store data.

Level -2 : Foreign Server used for authentication the store data.

We are going to generate public key and private key to store data secure.

In our novel infrastructure for Multiserver environment, we resolve the conflict between security and scalability by constructing a two-level authentication architecture and introducing the idea of public-key Cryptosystems. Obviously, our generic framework consists two levels: the first is single-server authentication and the second level is the trust chain among user, the "mother" server and "foreign" server. When the user authenticates with the "mother" server, it just acts like single-server authentication and with the help of the "mother" server, the user is able to perform authentication with the "foreign" server. Compared with the traditional framework, this infrastructure enables the existing single-server authentication protocols to convert to MA protocols.

In this paper, We demonstrate that one of the additive homomorphic encryption algorithm, the EC-OU(Elliptic curve Okamoto-Uchiyama) algorithm performs better than other additive homomorphic encryption algorithms.

B. Proposed Methodology

One way to compute functionality is to use a trusted third party. In the case of a trusted third party; each party sends their data to the TTP, which then computes the result of the data and sends the output to everyone. However, the key problem of cryptography is to achieve a truly trusted TTP. This requires a protocol that eliminates TTP.

In this paper, we propose a protocol that eliminates TTP and minimizes the overhead associated with cryptography-based privacy-preserving methods. There are 3 ways to perform functions in SMC. Unintentional transport protocols, homomorphic encryption, and secret sharing. Due to its high computational and communication costs, unintentional transmission based methods are not scalable, so we do not consider this method.

In the secret sharing method, a private channel needs to exist. But communication here is expensive due to the exchange of messages between parties. In the secret sharing method, the secret is divided into n s1-sn and one is given to each party.

Our secure multiparty addition protocol achieve better performance in terms of communication cost as compared to corresponding secret sharing based approach and hence scalable with respect to a number of parties.

C. Proposed Architecture:

Cryptography can be divided into three different types:

- Key cryptography
- Public key cryptography
- Hash function

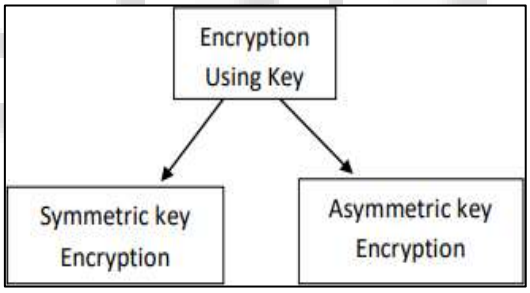

Fig. 1: Encryption Design architecture

The above Figure 1, shows the Encryption Design architecture. This has two types of encryption keys, one is Symmetric key Encryption and other is Asymmetric key Encryption. Key cryptography or symmetric cryptography uses a single key to encrypt data. Both encryption and decryption in symmetric cryptography use the same key, which makes it the simplest form of cryptography.

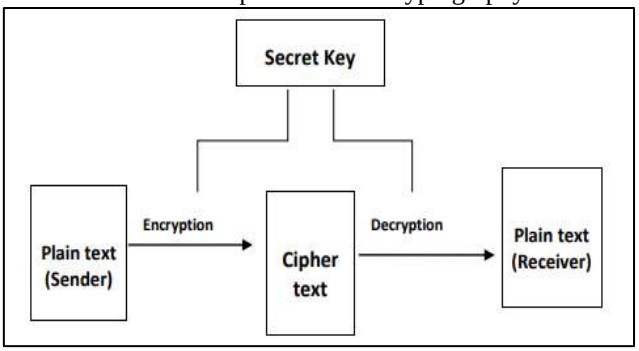

Fig. 2: Key Generation Design Architecture

From the above figure 2, we can clearly understand the Key Generation Design architecture

- Step 1: Secret key is used to both encryption and decryption of data and the data is shared between the receiver and sender of encrypted data.
- Step 2: Encryption provides confidentiality of data by transforming the Plaintext (Sender) into Ciphertext.
- Step 3: Decryption transforms Ciphertext back to Plain text (Receiver). It is generally a reverse process of Encryption.

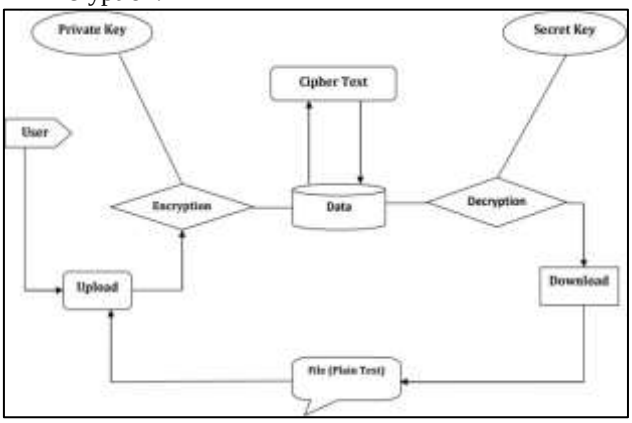

Fig. 3: Proposed Architecture

From the above figure 3, we can clearly observe the entire process of our proposed methodology. It can clearly explains that how the data is transformed from Cipher text to Plain text.

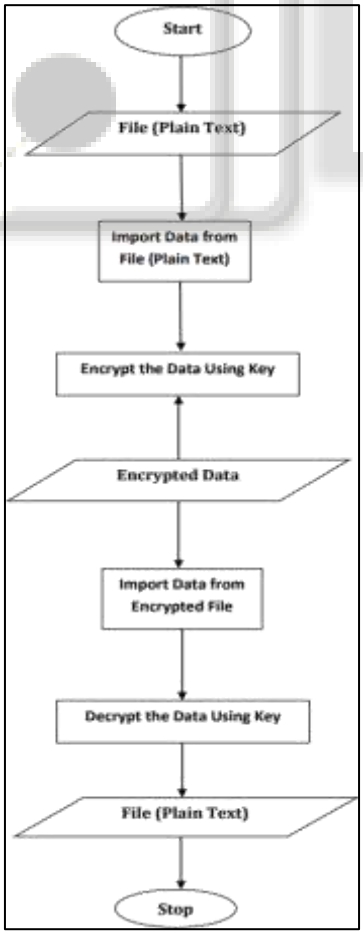

Fig. 4: Flow Chart Diagram for the Encryption and Decryption Process

IV. IMPLEMENTATION

When it comes to securing the actual data managed in an embedded device, cryptographic algorithms are one of the most reliable implementations of security through the middleware layer. Cryptographic schemes utilize some combination of encryption keys, obfuscator tools, digital signatures, and/or certificates, to name a few. This allows the sender to perform some type of encryption on the data before transmission to help ensure that "only" the "intended" recipient can decrypt the data. Again, here the core middleware with embedded JVM implementation can be used as a base, including some Java-based APIs for cryptographic support.

Algorithms such as AES, DSA, DES, SHA, PKCS#5, and RC4, to name a few, asymmetric and symmetric cryptographic digital signature key generators and factory message authentication code message digests. To illustrate this difference, let's consider the classic case used by most banks, where once the server is authenticated and the TLS tunnel is established, the user is authenticated with a password. Eavesdropping attacks would not be possible in this case, and brute force or dictionary attacks would also be ineffective in terms of identity theft due to the limited number of authorized authentication attempts. However, phishing attacks work well in this case: unsuspecting users don't think about checking if a TLS session has been established, so it's easy to give their details to the attacker.

A. Confidentiality:

Data confidentiality is an important security prerequisite for IoT networks. In a network, nodes require confidentiality of data during transmission. Data confidentiality means guaranteeing that data can only be accessed and modified by authorized entities. Not only users, but authorized things can also access information, which requires access control methods and authentication processes of the device (Miorandi et al., 2012; Sankaran, 2016). It protects data from passive attackers, keeping any data sent over an IoT network confidential. Providing confidential data is critical for IoT applications. To provide data confidentiality in the network, cryptographic algorithms are generally preferred over encrypted data. This way, even if the exchanged data is heard, the attacker cannot access its content.

B. Availability:

IoT environments contain sensors that provide important services. Because these IoT services can be accessed from anywhere at any time to provide continuous data, it is impossible to satisfy this property using a single security protocol. There can be many strategic steps to confirm availability.

C. Mathematical Function:

function encrypt(sequence):
L <-- left part of sequence (32 bit)
R <-- right part of sequence (32 bit)
R1 <-- apply expansion PBox on R
// apply non-invertible function

R2 <-- apply non-invertible modular functionon R1 using key K_i

// apply substitution boxes on R2 (48 bit -->32 bit)

R3 <-- apply 8 substitution boxes on R2

// apply XOR (^) on left and rightL1 <-- L ^ R3
// return combined resultreturn L1 + R3
endfunction

// same steps as encryptionfunction decrypt(sequence):
return encrypt(sequence)endfunction

V. ALGORITHM

We will use an additive full-state encryption scheme for evaluation. We consider four additive homomorphic encryption algorithms. They are Elliptic Curve Naccache-Stern (EC-NS), Elliptic Curve Okamoto-Uchiyama (EC-OU), Elliptic Curve Paillier (EC-P) and Elliptic Curve ELGamal (EC-EG) encryption. We demonstrate that the EC-OU algorithm performs better.

A. EC-OU algorithm:

The EC-OU algorithm utilizes elliptic curves in which both variables and coefficients are confined to elements of a finite field.
Input:
k: Private key (integer)n: Order of the curve
M: Message to be signed
Output:
(r,s): The signature
Generate a random key Q=(k,(x,y)).
Convert x to the integer j. This step is omittedfor prime curves since x is already an integer.
r = j mod n, if r=0, go to step 1.
H = hash(M)
Truncate H to the leftmost ceil(log2n) bits.
Convert H to an integer e by loading it in bigendian fashion.
Compute s = k(e+kr) mod n; if s=0, go to step1.
Return (r,s).

Fig. 5: Files Upload to Cloud

From the figure 5, we can upload the data file to cloud.

Fig. 6: Encrypted File

From the figure 6, we can observe that the uploaded file is converted into encrypted file which is a non-readable format.

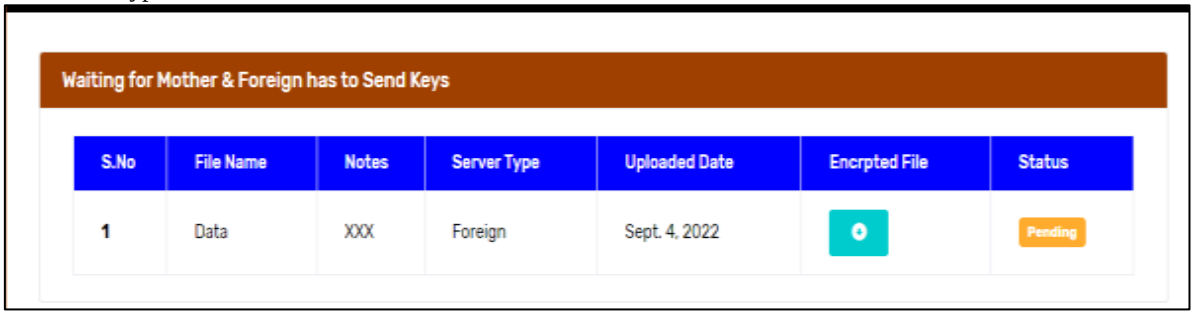

Fig. 7: Waiting for Mother and Foreign has to send Keys

From the figure 7, we can observe that the data is waiting for both mother server and foreign server has to send public and public keys.

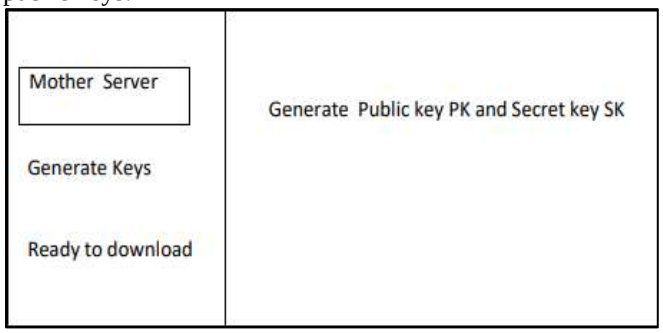

Fig. 8: Keys Generated

From the figure 8, we can clearly identify that both the Public key(Pk) and the Secret key(Sk) are generated.

Fig. 9: Keys Generated and Ready to Download

From the figure 9, we can observe that both the Public key and the Secret key are generated and ready to download the decrypted file.

Fig. 10: View Secret and Public Keys

From the figure 10, we can view the Mother Secret key and Public key as well as the Foreign Secret key and Public key.

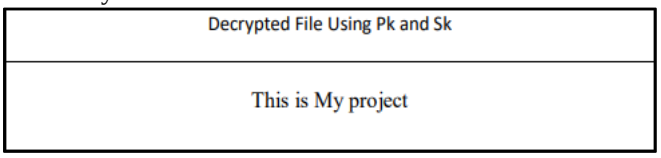

Fig. 11: Decrypted File

From the figure 11, we can see that the uploaded file is transformed into decrypted file, which is a readable format by using the Public key(Pk) and the Secret key(Sk).

VI. CONCLUSION

In this paper, we propose a novel approach to cryptographically secure multi-party computation using elliptic curves. We experimentally evaluate various ECC-based homomorphic encryption schemes for our proposed protocol. We demonstrate that the EC-OU algorithm performs better among the four selected algorithms. Compared to the corresponding secret-sharing- based methods, our secure multi-party addition protocol achieves better performance in terms of communication cost and is therefore scalable to multi-party. Therefore, in future research, we will use EC-OU for privacy protection in data mining applications.

REFERENCES

[1] O.Goldreich, "The Foundation of Cryptography," Vol.2. Cambridge Univ. Press, Cambridge, 2004.
[2] Y.Lindell and B.Pinkas,"Secure Multiparty Computation for Privacy – preserving Datamining, "Journal of parivacy and confidentiality, Vol.1, No.1, 2009, pp.59-98.
[3] Sankitaj Patel, Ankit chaouhan, Devesh C. Jinwala" Comparative Evaluation of ECC based homomorphic encryption schemes for a novel Secure multiparty computation", Journal of information Security, Jan 2014.
[4] Swathi N S, SwathiP,Rajesh N V "Efficient Performance Analysis of Elliptic Curve Cryptography over RSA to Secure the Data",International Journal of Computer & Mathematical Science , Volume 7,2018R.
[5] FaiqaMaqsood, Muhammad Mumtaz Ali, Muhammad Ahmed, Munam Ali shah,"Cryptography: A Comparative Analysis for Modern Techniques", International Journal of Advanced Computer Science and Application, 2017.
[6] Aquino ValentimMota, Sami Azam, Kheng Cher Yeo "Comparative Analysis of Different Techniques of Encryption fr Secured Data Transmission" IEEE international conference on power ,control, signals and Instrumentation engineering, 2017
[7] Patil Maulik Y, Manjusha Yeola"Generating Private Recommendation Using ELGamal Homomorphic Encryption" IJECS, 2015.

[8] XianjinFang,Yanting Wu "Investing into the Elliptic curve cryptography"3rd International conference on Information Management,2017.
[9] Aquino ValentimMota,SamiAzam,Kheng Cher Yeo"Comparative analysis of different techniques of encryption for Secured Data transmission" IEEE International conference on power, control, signals and instrumentation engineering,2017.
[10] V SudarshanRao, Nsatayanarayan "Secure and Practical outsourcing of linear programming in cloud computing ", International Journal of computer applications, 2017.
[11] Moncef Amara,and Amar Siad,"Elliptic curve Cryptography and its application ,"International workshop on system ,signal processing and application .IEEE,2011
[12] Non Thiranant, Young Sil Lee, HoonJae Lee, "Performance comparison between ECC and RSA, "International Conference of Advanced Information Networking and Application,2015
[13] Sankita patel,Sweta Garasia and Devesh Jinwala ,"An efficient Approach for privacy preserving distributed K-means clustering based on shamir secret shring scheme ",International federation of information processing ,2012.
[14] Payal.V.parmar, Sharddha B Padhar, Shafika N .patel, Niyatee I.Bhatt, Rutvij H.Jhaveri,"Survey of various Homomorphic Encryption algorithm and schemes"Vol.91. International journal of computer applications, April 2014.

