ISSN (e): 2250 – 3005 || Volume, 13 || Issue, 4|| Jul.-Aug. – 2023 ||
International Journal of Computational Engineering Research (IJCER)

Enhancing Secure Multiparty Computation Through Exploration of Elliptic Curve Cryptography-Based Homomorphic Encryption Algorithms

Madhira Srinivas[1], Porika Sammulal[2]
[1]Research Scholar,
Department of Computer Science & Engineering,
JNTUH University, Hyderabad, Telangana
[2]Professor,
Department of Computer Science & Engineering,
JNTUH University, Hyderabad, Telangana
Corresponding Author: Madhira Srinivas

Abstract:
In this research work, we investigate an Elliptic Curve Cryptography-based approach to Secure Multiparty Computation (SMC). The extensive availability of data and the advancement of communication technologies have made it feasible for people in a scattered settings to collaborate on computations. The confidentiality of the information shared between the parties is crucial. The conventional approach to SMC is to have a trusted third party (TTP) do the calculations. TTPs are difficult to implement in practice, which is why eliminating them from SMC is so important. Traditional homomorphic encryption methods, such as RSA and Paillier, are also used in the current proposed solutions for SMC. The increased costs of utilizing such cryptosystems render the resultant SMC protocols unscalable. We propose an Elliptic Curve Cryptography (ECC)-based approach to SMC that is TTP-free and scalable (in terms of computational and communication cost). Research and study of the many ECC-based homomorphic schemes described in the literature is required in order to choose the most suitable scheme for a given application. We conduct an empirical analysis of the available ECC-based homomorphic encryption algorithms, contrasting their relative merits in terms of metrics like computational and transmission overhead. From a pool of viable options, we advise one that is both secure and efficient enough to be utilized for any work that has to be kept secret.

Keywords: *Elliptic Curve Cryptography (ECC), Secure Multiparty Computation (SMC), a trusted third party (TTP), homomorphic encryption, computationaloverhead, transmission overhead*

Date of Submission: 18-08-2023 Date of acceptance: 04-09-2023

I. Introduction

In the last few decades, data privacy and security has become the primary concern of everyone. Due to the rise in technological advancements and the internet, it has been a challenging task to provide data security and data privacy of the data, when data is distributed over large distributed networks. As everyone is now concerned with their data, a lot of research is going on how to provide data security and privacy to the participants in the network. One of the techniques that provide the solution to the problems of data security and data privacy is Secure Multiparty Computation.The secure multiparty computation may be defined as the problem of 'n' players computing jointly on an agreed function securely on the inputs without revealing them. Our handhelds have become smaller; computers faster; disks larger; networks more efficient and we enjoy bandwidths like never before; everything has grown exponentially. All this sums up to a very favorable environment for data collection, transfer, and storage. In order to fully utilize this data, there is a need to perform collaborative computation on data. However, the data collected mostly contain information related to individuals, their financial status, lifestyle, and social behavior in general. Joint computation of data may pose a threat to the privacy of individual data. Hence, there is a need to protocol a device that performs joint computation on private data without revealing data to other parties. Secure Multiparty computation (SMC) addresses this issue. The general framework for SMC consists of specifying a random process that maps m inputs (local inputs of parties) to m outputs (desired outputs) [1]. The random process describes the desired functionality. We focus on the addition function in this paper. There are many real-world scenarios where privacy can be an issue, a few of which are worth mentioning: Considering the field of medical research, considering the case that a number of different hospitals wish to perform joint research on their patient data. Also, let us assume that privacy policy and law prevent these hospitals from over pooling their data or revealing it to each other, due to the confidentiality of patient records. In such cases, it is necessary to find a solution that enables the hospitalsto compute the desired functionality on the union of their databases, without ever pooling or revealing their data. Consider the interaction between different intelligence agencies; for security purposes, these agencies cannot allow each other free access to their confidential information; if they did, then a single mole in a single agency would have access to an overwhelming number of sources. It is much more likely that suspicious behavior would be detected if different agencies were able to perform computations on their combined data [2]. One way to compute the desired functionality is to use Trusted Third Party (TTP). In this scenario, parties send their data to TTP and TTP then computes the results on parties' data and sends the output to all parties. However, the pivotal question in cryptography is to achieve TTPs that are indeed trusted. This demands protocols that eliminate TTP. In this paper, we propose a protocol that eliminates TTP. There are three approaches to performing desired functionality in Secure Multiparty Computation viz. the Oblivious Transfer [3] Protocol, the homomorphic encryption [4], and the secret sharing [5]. The oblivious transfer protocol is costly in terms of computational and communication overheads. The secret sharing-based approach gives better results in terms of computational cost due to the primitive operations involved [6]. However, it requires the existence of private channels. In addition, as pointed out in [7], the communication cost is higher due to message exchange between other parties in the protocol as explained in Section 3. The homomorphic encryption-based approach does not require the existence of a private channel and assures a high level of privacy. Hence, we focus on a homomorphic encryption-based approach in this paper. The homomorphic encryption is introduced in Section 2.

There are methods described in the literature for carrying out safe multiparty computing through homomorphic encryption. However, they rely on age-old methods of encryption [8,9]. As with traditional encryption methods, Elliptic Curve Cryptography (ECC)-based approaches show promise [10,11]. This is because ECC offers more protection for each individual bit. As an example, an ECC technique requires 160 bits of parameters in order to achieve the same level of security as 1024-bit RSA [12]. In this research, we use an ECC-based strategy to develop a multi-party computing protocol that is both secure and efficient. Section 3 explains the methodology. In Section 4, we conduct an empirical analysis of many different additively homomorphic encryption systems.

Some methods use ECC-based encryption systems. However, they are computationally costly [13,14] because of the need for repeated encryption decryption at each location. Our method minimizes the computational burden of SMC by eliminating the need for numerous cipher operations at each location. In Section 3 we explain why.

II. Background and Related Work

Wang et al. [15] developed an efficient privacy preservation mechanism for securing the data stored in the cloud. Here, the Public Key Encryption with Fine Grained Searchable Capability (PEFKS) mechanism was utilizedwith fine-grained searchable capability. The securityproperties analyzed in this work were confidential property,fine-grained searchable property efficiency property, and privacy-preserving property. The limitation behind thiswork was, it used to reduce the computational complexityof this mechanism. Cao et al. [16] suggested a multi-keyword ranked search algorithm for increasing the privacy preservation of cloud data. The main intention of thiswork was to safeguard the privacy of sensitive data before outsourcing it to the users. Moreover, the problem of multi-keyword search was investigated in this paper, which established the details about securing the data under some privacy requirements. Also, the measure of coordinate matching was utilized to estimate the similarity between two different threat models. The major advantages studied in this paper were reduced computation and communication overhead. Still, it required exploring some other multi-keyword semantics to ensure the integrity of the data. Hayward and Chiang [17] implemented a parallelizing fully Homomorphic encryption mechanism for protecting the privacy of stored data in the cloud. Here, the Gentry's encryption algorithm was used to speed up the performance of encryption. Dhote [18] suggested a fully homomorphic encryption (FHE) technique to secure the cloud data. Here, different types of security issues were analyzed, which includes availability, third-party control, legal issues, and privacy. This technique allowed the users to perform various operations on the encrypted data. Also, it aimed to improve the confidentiality of the data with a reduced number of computational steps. However, it required to reduce the size of cipher text for efficiently processing the data. El Makkaouri et al. [19] developed a fast cloud Rivest Shamir Adelman (RSA) algorithm for increasing the confidentiality of the data. This encryption mechanism contains four components, which includes key generation, encryption, evaluation, and decryption. The benefit of this mechanism was. It provided a good performance with reduced running time. Dasgupta and Pal [20] developed a symmetric Homomorphic encryption scheme for performing arbitrary operations on the data. Farokhi et al. [21] suggested a semi-homomorphic encryption mechanism for increasing the security and privacy of cloud data. Here, the bilinear encryption algorithm was utilized to ensure the closed-loop system stability, which ensured the bounds of the closed-loop performance. The steps involved in this work were as follows: key generation, encryption, and decryption. Also, the discrete-time linear time-variant system was implemented to develop the control architecture. The closed-loop system stability was guaranteed to increase the closed-loop system performance. Still, this work required to implement the dynamic feedback controllers to make the computation faster. Kaaniche and Laurent [22] investigated various privacy and security issues related to cloud data storage. Here, some cryptographic defense mechanisms were surveyed to address the protection of the outsourced data.

Yao [23] first described the multi-party computation issue, and Goldreich, Micali, and Wig- derson [24] expanded on it. The primary technique they use is modeling the issue as a combinatorial circuit. Each gate in the circuit is then processed by a participant-run procedure. Despite being broad and straightforward, the method is not scalable for big inputs because of combinatorial explosion in circuit size. Goldreich [1] proposes a number of different ways for doing multiparty computing, such as homomorphic encryption and secret sharing, in addition to the scrambled circuit. Privacy-Preserving Data Mining [25,26], Privacy-Preserving Statistical Information Retrieval [27,28], and Privacy-Preserving Database Access [29] are just few of the applications that have been presented in the literature that call for Secure Multiparty Computation.

Different classification [30], clustering [31], and association rule mining [32] strategies have been presented in Privacy Preserving Data Mining. Structure's bare bones private value adding and limited transparency as the stumbling comparison. The emphasis of this study is on private value advertising. If we take the case of privacy-preserving distributions, for clustering using weighted average probabilities lem(WAP) as a fundamental element [33]. To the Partieslocal clustering at each location, then worldwide WAP is used to get the average of each cluster. Case in point Take into account a case involving two parties, A and B. In total, for a particular cluster, and the total amount of data after. The sumA and sumB are the local cluster means for A and B, respectivelynA and nB as the total of nA and nB.

2.1 Homomorphic Encryption:

Simple calculations may be performed on encrypted data using Homomorphic Encryption methods. Due to the fact that the calculations are done on encrypted data, confidentiality is maintained. The encrypted sum or encrypted product of two encrypted communications is usually something a third party can figure out.

This property of homomorphic cryptosystems makes them applicable to many privacy-preserving protocols for safe multiparty computing. The public key of the encryption system, together with two cipher-texts, are fed into an additive homomorphic algorithm, and the resulting ciphertexts are deciphered.

$E_{P_k}(m_1) +_{P_k} E_{P_k}(m_2) = E_{P_k}(m_1 + m_2)$; where $+_{P_k}$is the homomorphic addition function, E_{P_k} is the public- k key encryption function $m_1 \, and \, m_2$are elements in the domain of data. We refer additively homomorphic encryption scheme based on ECC in this paper.

2.2 Elliptic Curve Cryptography (ECC)

ECC is a public key cryptography approach based on the algebraic structure of elliptic curves over finite fields [10,11]. There are two types of finite fields where the elliptic curves are defined: prime fields Fp, where p is a large prime number, and binary fields F_2^m. In this work, we are interested in the use of elliptic curves over prime fields E (Fp). A nonsupersingular elliptic curve E over Fp is defined as the solution of (x ,y) ∈ (F_p × F_p) to the cubic equation:

$y^2 = x^3+ax + b \bmod p$

where a, b ∈F_p, such that $4a^3 + 27b^2 \neq 0 \pmod p$ to- gether with a special point ∞ called the point at infinity, The group of points forms an abelian group with addition operation so that the addition of any two points results in another point on the same curve. ECC-based cryptographic protocol security is based on the Elliptic Curve Discrete Logarithm Problem (ECDLP). The ECDLP can be defined as the problem of finding the sca- lar k such that Q = kP given Q and P (generator point).

III. Algorithms Used for Evaluation:

We utilize additively homomorphic encryption schemes for evaluation. We investigate four algorithms that are additively homomorphic. They are Elliptic Curve Naccache-Stern (EC-NS) Encryption [26], Elliptic Curve Okamoto-Uchiyama (EC-OU) Encryption [27], Elliptic Curve Paillier (EC-P) Encryption [28] and Elliptic Curve

Elgamal(EC-EG) Encryption [29]. First three schemes are described in [30] and are Elliptic Curve variants of the previously proposed schemes by Naccache-Stern [26],
Okamoto-Uchiyama [27] and Pailler [28]. EC-EG is thevariant of ElGamal [29]. Original ElGamal is multiplicatively homomorphic while EC-EG is transformed to additive group and hence is additively homomorphic. The pseudo code and other details may be found in [30].

The Proposed Approach: Secure Multiparty Computation Using ECC

We propose a novel approach to the Secure Multiparty Addition problem using ECC. Secure Multiparty Addition has been implemented for privacy-preserving data mining using homomorphic encryption [25] and secret sharing [7]. However, the approach of [25] uses classical public key encryption scheme and hence is computationally expensive. Further, multiple cipher operations at each site increased the computational cost. The secret sharing-based approach proposed in [7] is efficient in terms of computational cost. However, two rounds of information exchange among parties increase the communication cost. Among the three approaches to implement SMC viz. the oblivious transfer based, the homomorphic encryption-based and the secret sharing based, we focus on homomorphic encryption based approach.

We allow parties to communicate in ring topology. Consider three party scenario with parties A, B and C with private values m1, m2 and m3 respectively. Parties need to compute m1+ m2 + m3 securely. One party in the protocol is randomly designated as the Initiator party. The proposed approach considering party A as the Initiator is shown in Figure 1.

As shown in Figure 1, the initiator first encrypts its private value using ECC based encryption scheme. The resultant ciphertext (which is in the form of an elliptic curve point) is sent to the next party in the ring. The next party does not perform any cipher operation but just adds its own private value (mapped to an elliptic curve point) with the received cipher text. This process is repeated and finally initiator receives the message E(m1) + m2 + m3 at the end of phase I. In phase II, the initiator decrypts the message by removing the noise (that was added during encryption) from the message and computes m1+ m2 + m3. Here, the initiator just removes the noise in order to get the desired sum. Hence, we have slightly modified the decryption processes of algorithms to get the sum value. This sum is then sent to next party in the ring and eventually all parties will receive the sum.

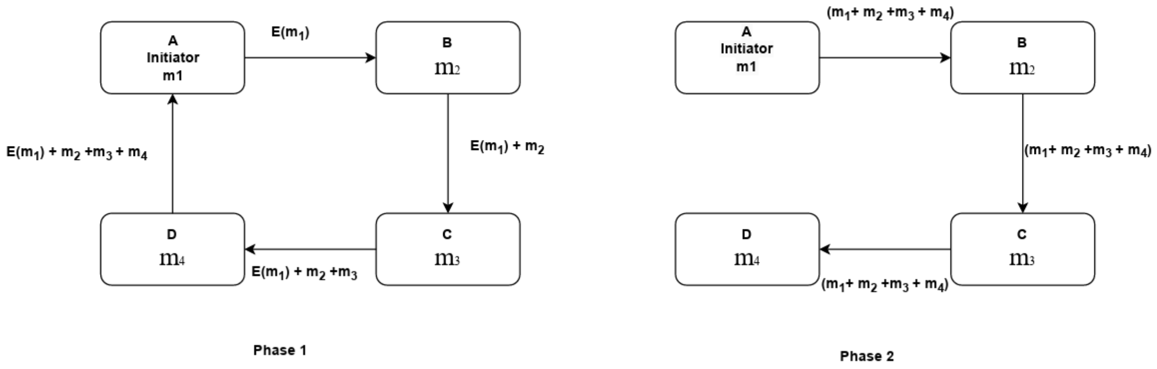

A
Initiator
m1
$E(m_1)$
B
m2
$E(m_1) + m_2$
C
m3
$E(m_1) + m_2 + m_3$
D
m4
$E(m_1) + m_2 + m_3 + m_4$
Phase 1
A
Initiator
m1
$(m_1 + m_2 + m_3 + m_4)$
B
m2
$(m_1 + m_2 + m_3 + m_4)$
C
m3
$(m_1 + m_2 + m_3 + m_4)$
D
m4
Phase 2

Figure 1. Secure Multiparty Addition using Elliptic Curve Cryptography

IV. Theoretical Analysis

In this section, we discuss the computational and communication costs of our proposed approach.

4.1. Computational Cost

The computational cost is mainly due to the operations in the first phase where parties send data to the next party in the ring. However, we need to consider the cost for the initiator and the rest of the parties separately. This is because the initiator encrypts and decrypts the data and the rest of the parties just need to add their data to the received data. Suppose to encrypt the data, the time taken is T1. It is pointed out in [10] that by means of O(logk) addition and doubling, one can compute k. M value. Hence, T1 becomes O(logk). Further, to decrypt the data, the time taken is T2. Let us have the time taken by other parties to add their data point is T3. Hence, the computation cost for the initiator can be = (O(logk) + T2) = O(logk). For the rest of the parties, the cost becomes T3 and hence O(1). For the total of N parties, the cost becomes O(log k) + O(N).

4.2. Communication Cost

Our proposed approach involves two phases. In the first phase, a single message (as elliptic curve points) is communicated by each party to the next party in the ring. Hence, total N messages are transmitted for the N party scenario. In the second phase, only the sum (that is calculated) by the initiator party is communicated to all parties in a ring. The last party in the ring need not send this sum to the initiator. Hence, total (N-1) messages are transmitted in the second round. Thus for N parties in the protocol, the communication cost becomes O(N) i.e. linear in terms of a number of parties.

4.3. Comparison with Secret Sharing-Based Approach

Our proposed approach is closest to the approach of [7]. In [7], a privacy-preserving clustering protocol is proposed using a secret sharing scheme. The basic building block they use is the secure multiparty addition. Hence, we compare our approach with the approach of [7]. The approach of [7] is efficient in terms of computational cost due to primitive operations required in secret sharing scheme. However, approach of [7] requires message exchange among every other party in the protocol. Table 1 shows the comparison in communication costs. We achieve O(N) complexity with respect to communication cost as compared to O(N2) of the secret sharing-based approach. Hence, our approach achieves scalability with respect to a number of parties in the distributed scenario.

	Encryption	Decryption	Total Complexity
Proposed Model	N	(N-1)	2N-1, hence O(N)
Existing Model	N(N-1)	N(N-1)	2N(N-1),hence O(N2)

V. Experimental Results

In this paper, we focus on a comparative evaluation of ECC-based homomorphic encryption schemes to implement secure multiparty addition. We implement our proposed secure multi-party addition protocol using four ECC-based schemes such as EC-NS, EC-OU, EC-P and EC-EG.

5.1. Experimental Setup

We carry out the experiments in Python. The experiments are conducted on three different machines to emulate the true distributed scenario consisting of three parties. All machines have similar configurations of Intel Core i5processor, 4GB of RAM, and 3.20 GHz of processing power. All reported results are averaged from 5 runs of the protocol. All the participating parties initially agreed upon the ECC parameters required for the respective ECC algorithm. The ring topology among the parties is set up and each party knows its neighbor on ring. One party in a ring is randomly designated as Initiator. The secure three party addition protocol is implemented with four different homomorphic algorithms based on ECC. Our test application successfully shows fully functional secure three party addition protocol over real network.

5.2. Experimental Results

We evaluate ECC based encryption schemes for secure multiparty addition based on two metrics viz. the computational cost and the communication cost. Computational cost is measured as time taken for computation and communication cost is measured as number of bytes exchanged over communication channel. We perform experiments with different ECC parameter sizes viz. 112-bit, 160-bit and 256-bit. For EC-OU, the prime value is 341 bit long and for remaining encryption schemes, prime value is similar to the size of Elliptic Curve Parameter Size i.e. 112 bit or 160 bit or 256 bit.

Table 2 shows the cost for running secure three party addition protocol using four ECC based encryption schemes. As shown in Table 2, EC-EG gives better performance among all in terms of computational cost. However, if we consider the communication cost, EC-OU gives better performance. The result for higher computational cost for EC-OU is that in EC-OU, prime number p,q are 341 bits long and for other schemes prime numbers are same as the size of EC parameter size. To get the fair comparison, we then took random parameter size as 341 bits for all the algorithms and measure the cost of running secure multiparty addition using 341-bit value. The results are shown in Table 3.

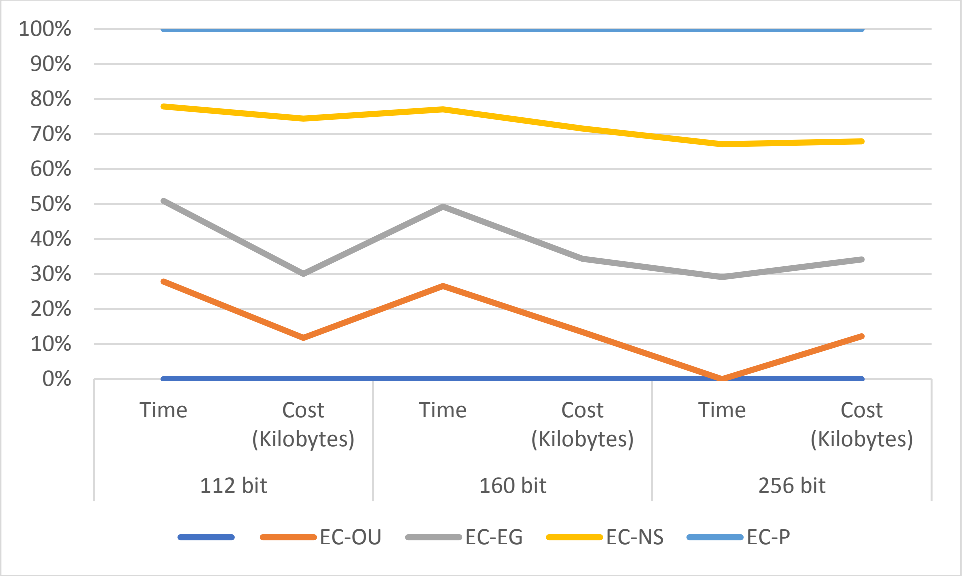
Table 3 shows that keeping 341-bit prime value fixed for all schemes results in higher computational cost for the encryption schemes other than EC-OU. In these experiments, we found that EC-OU and EC-EG takes lesser time where as EC-OU takes lesser space which is reasonably lesser than the other three schemes. Results for computational and communication cost are shown in Figures 2 and 3 respectively for varying length of primes. Results for computational and communication cost for 341-bit fixed length prime are shown in Figures 4 and 5 respectively. As shown in Figures 4 and 5, if we consid- er both computational and communication cost, EC-OU perform better than all algorithms selected for evaluation.

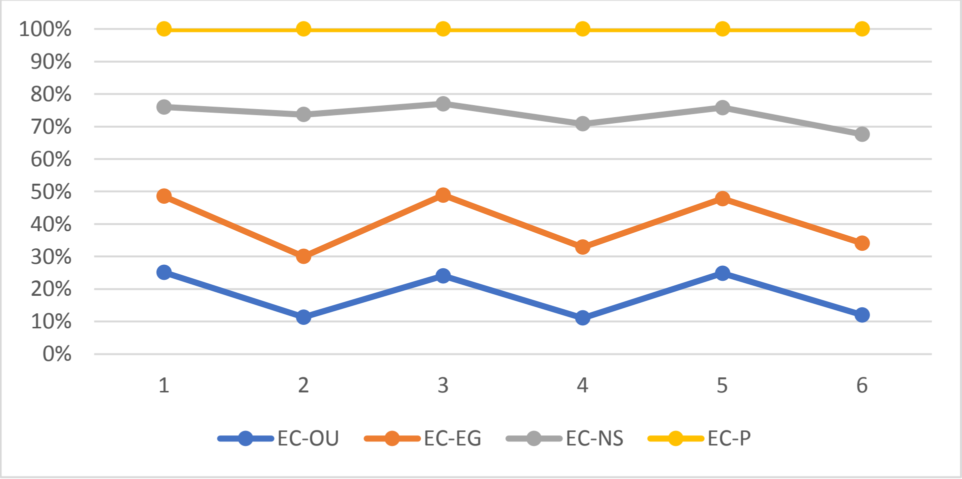
Table 2.a. Results for Secure Multiparty addition using various ECC-based Encryption Schemes

Algorithm	112 bit		160 bit		256 bit	
	Time (Millli Sec.)	Cost (Kilobytes)	Time (Millli Sec.)	Cost (Kilobytes)	Time (MillliSec.)	Cost (Kilobytes)
EC-OU	311	1117	334	1760	400	2720
EC-EG	256	1739	285	2763	310	4866
EC-NS	301	4220	350	4907	403	7486
EC-P	247	2438	289	3740	350	7124

Table 2. b. Results for Secure Multiparty addition using various ECC-based Encryption with 341 primes in all algorithms

Algorithm	112 bit		160 bit		256 bit	
	Time (Millli Sec.)	Cost (Kilobytes)	Time (Millli Sec.)	Cost (Kilobytes)	Time (MillliSec.)	Cost (Kilobytes)
EC-OU	311	1117	334	1460	400	2720
EC-EG	290	1839	345	2863	370	4987
EC-NS	340	4300	390	4999	450	7580
EC-P	297	2589	320	3840	390	7324





VI. Conclusion:

In this paper, we propose a novel approach to securing multiparty computation using elliptic curve cryptography. We empirically evaluated various ECC-basedhomomorphic encryption schemes for our proposed protocol. We demonstrate that the EC-OU algorithm performs better among the four selected algorithms. Our secure multiparty addition protocol achieves better efficiency in terms of communication cost as compared to the corresponding secret-sharing-based approach and hence is scalable with respect to a number of parties. In addition, we highlighted various applications such as privacy-preserving data mining as the candidate applications for our proposed approaches. Our future includes incorporating EC-OU-based secure multiparty computation in privacy preserving data mining application.

References:

[1]. O. Goldreich, “The Foundations of Cryptography,” Vol. 2. Cambridge Univ. Press, Cambridge, 2004. Y. Lindell and B. Pinkas, “Secure Multiparty Computation for Privacy-Preserving Data Mining,” Journal of Privacy and Confidentiality, Vol. 1, No. 1, 2009, pp. 59-98.

[2]. M. Rabin, “How to Exchange Secrets by Oblivious Transfer,” Technical Report Tech. Memo TR-81, Aiken Computation Laboratory, 1981. D. Josep Ferrer, “A new privacy homomorphism and applications,” Information Processing Letters, Vol. 60, No. 5, 1996, pp. 277-282. http://dx.doi.org/10.1016/S0020-0190(96)00170-6

[3]. A. Shamir, “How to Share a Secret,” Communication of the ACM, Vol. 22, No. 11, 1979, pp. 612-613. http://dx.doi.org/10.1145/359168.359176 T. B. Pedersen, Y. Saygin and E. Savas, “Secret Sharing vs. Encryption-Based Techniques for Privacy-Preserving Data Mining,” UNECE/Eurostat Work Session on SDC, 2007.

[4]. S. Patel, S. Garasia and D. Jinwala, “An Efficient Approach for Privacy Preserving Distributed K-Means Clustering using Shamir’s Secret Sharing Scheme,” In T. Dimitrakos, R. Moona and D. Patel, Eds., Trust Management VI, IFIP Advances in Information and Communication Technology, Vol. 347, Springer, Boston, 2012, pp. 129-144.

[5]. G. Jagannathan and R. N. Wright, “Privacy-Preserving Distributed k-Means Clustering over Arbitrarily Partitioned Data,” KDD, ACM Press, 2005, pp. 593-599. S. Jha, L. Kruger, and P. McDaniel, “Privacy-Preserving Clustering,” 10th European Symposium on Research in Computer Security, 2005, pp. 397-417.

[6]. N. Koblitz, “Elliptic Curve Cryptosystems,” Mathematics of Computation, Vol. 48, 1987, pp. 203-209. http://dx.doi.org/10.1090/S0025-5718-1987-0866109-5

[7]. V. S. Miller, “Use of Elliptic Curve in Cryptography,” In Proceedings of Advances in Cryptology (CRYPTO’85), Springer Verlag, 1986, pp. 417-426.

[8]. Certicom Research, “Standards for Efficient Cryptography—SEC 1: Elliptic Curve Cryptography,” 2009.

[9]. A. C. Patel, U. P. Rao and D. R. Patel, “Privacy-Preserving Association Rules in Unsecured Distributed Environment Using Elliptic Curve Cryptography,” Proceedings of International Conference on Computing Communication & Networking Technologies (ICCCNT), 2012, pp. 1-5.

[10]. M. Rajalakshmi and T. Purusothaman, “Privacy-Preserving Distributed Data Mining using Randomized Site Selection,” European Journal of Scientific Research, Vol.64, No. 4, 2011, pp. 610-624.

[11]. Wang, X.A., et al.: Efficient privacy preserving predicate encryption with fine-grained searchable capability for Cloud storage. Comput. Electr. Eng. 56, 871–883 (2016)

[12]. Cao, N., et al.: Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Trans. Parallel Distrib. Syst. 25, 222–233 (2014)

[13]. Hayward, R., Chiang, C.-C.: Parallelizing fully homomorphic encryption for a cloud environment. J. Appl. Res. Technol. 13, 245–252 (2015)

[14]. Dhote, C.: Homomorphic encryption for security of cloud data. Procedia Comput. Sci. 79, 175–181 (2016)

[15]. El Makkaoui, K., et al.: Fast cloud-RSA scheme for promoting data confidentiality in the cloud computing. Procedia Comput. Sci. 113, 33–40 (2017)

[16]. Dasgupta, S., Pal, S.: Design of a polynomial ring based symmetric homomorphic encryption scheme. Perspect. Sci. 8, 692–695 (2016)

[17]. Farokhi, F., et al.: Secure and private cloud-based control using semi-homomorphic encryption. IFAC-PapersOnLine 49, 163–168 (2016)

[18]. Kaaniche, N., Laurent, M.: Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms. Comput. Commun. 111, 120–141 (2017)

[19]. A. C. Yao, “Protocols for Secure Computations,” Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science, 1982, pp. 160-164.

[20]. O. Goldreich, S. Micali and A. Wigderson, “How to Play any Mental Game,” Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 218-229.

[21]. B. Pinkas, “Privacy-Preserving Data Mining,” Journal of Cryptology, Vol. 50, No. 3, 2002, pp. 177-206. http://dx.doi.org/10.1007/s00145-001-0019-2

[22]. B. Pinkas, “Cryptographic Techniques for PrivacyPre- serving Data Mining,” SIGKDD Explorations Newsletter, Vol. 4, No. 2, 2002, pp. 12-19. http://doi.acm.org/10.1145/772862.772865.

[23]. W. L. Du and M. J. Atallah, “Privacy-Preserving Cooperative Scientific Computations,” 14th IEEE Computer Security Foundations Workshop, Nova Scotia, 11-13 June 2001, pp. 273-282.

[24]. W. L. Du and M. J. Atallah, “Privacy-Preserving Statistical Analysis,” Proceedings of the 17th Annual Computer Security Applications Conference, New Orleans, 10-14 December 2001, pp. 102-110.

[25]. W. L. Du and M. J. Atallah, “Protocols for Secure Re- mote Database Access with Approximate Matching,” 7th ACM Conference on Computer and Communications Security (ACMCCS 2000), The First Workshop on Security and Privacy in E-Commerce, Athens, 1-4 November 2000, pp. 87-111.

[26]. J. Vaidya and C. Clifton, “Privacy-Preserving k-Means Clustering over Vertically Partitioned Data,” Proceedings of 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, 2003, pp. 205-216.

[27]. J. Vaidya and C. Clifton, “Privacy-Preserving Association Rule Mining in Vertically Partitioned Data,” 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 639-644.

[28]. S. Jha, L. Kruger, and P. McDaniel, “Privacy-Preserving Clustering,” Proceedings of 10th European Symposium on Research in Computer Security, 2005, pp. 397-417.

[29]. D. Naccache and J. Stern, “A New Public Key Cryptosystem Based on Higher Residues,” ACM Conference on Computer and Communications Security, 1998, pp. 59- 66.

[30]. T. Okamoto and S. Uchiyama, “A New Public-key Cryptosystem as Secure as Factoring,” EUROCRYPT, 1998, pp. 308-318.

[31]. P. Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” EUROCRYPT, 1999, pp. 223-238.

[32]. T. ElGamal, “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” CRYPTO, IT, Vol. 31, No. 4, 1985, pp. 469-472.

[33]. P. Paillier, “Trapdooring Discrete Logarithms on Elliptic Curves over Rings,” ASIACRYPT, 2000, pp. 573-584.