Analysis of Security and Privacy in Secure Multiparty Computation

Deepthy Jose
Assistant Professor, Mercy College, Palakkad, Kerala, India

***Abstract*—Data security and privacy were the important concern related to distributed network. Secure multiparty computation (SMPC) plays an important role in security and privacy. Basically, this is a cryptographic technique and has started from 1970's based on some secure function evaluation methods. This paper is regarding the analysis of the various studies about the secure evaluation methods and encryption methods associated with Secure multiparty computation till this era. Emerging technologies like blockchain, mobile computing, quantum computing and cloud computing have resulted in the rebirth of secure multiparty computation. Various sectors like healthcare, finance, or the areas where the personal information were shared, plays a crucial role for the privacy and security terms. This paper analyzes how this method is achieved through this secure multiparty computation.**

***Index Terms*— block chain, cloud computing, mobile computing, SMPC, Secure multiparty computation.**

I. INTRODUCTION

Secure multiparty is a cryptographic approach where n parties can securely compute a function together without revealing their individual data inputs. This is for preserving data privacy and security while sharing the data in distributed systems. This method was originally coined in 1970's and revised after the introducing secure function evaluation. SMPC is utilized when privacy is crucial, such as financial transactions, healthcare data or personal information transferred between several entities. This paper aims to find out the analysis of SMPC techniques introduced and the case studies of techniques related to SMPC. The paper introduces about SMPC, secure function evaluation, secret sharing, and cryptographic primitives. And covers the hybrid method supported by SMPC like differential privacy method and homomorphic function methods.

Fundamental Principles and Techniques:

1. Secret sharing: This technique divides a secret value into shares dispersed among parties so that no subset of participants below a threshold can reconstruct the original data. This is one of the fundamental primitives of SMPC. Reconstruction the secret data takes place when the necessary number of parties collaborate [1]. For example, secret sharing takes a secret 's' and splitting it into multiple shares s_1, s_2, s_n and distributing shares among parties P_i. for the reconstruction of secret 's', the parties must collaborate and each bringing their respective share into the collective pool [2]. Another type of secret sharing method is Shamir's secret sharing, it is a cryptographic method that splits a secret into multiple pieces called shares distributing them among group of parties, where a predetermined minimum number of shares must be combined to reconstruct the initial secret. It is based on the polynomial interpolation method, for example a subset of t or more shares can reconstruct the secret but it revels less than t shares.

2. Oblivious Transfer: It is a basic primitive involves two parties, sender and the receiver, where the sender has n messages and the receiver wants to receive m of those messages with some conditions, that, sender does not learn which of the m message was received by the receiver, and the receiver only learns the content of the m messages they received. They learn nothing about the content of the other n-m messages they did not receive [3].

3. Garbled Circuits: These are important cryptographic primitive introduced by Yao for secure two-party computation [4]. In this technique multiple parties compute a function on private inputs without revealing and transforming the function into an encrypted garbled form that can be evaluated securely. It

contains a circuit representation where the function is to be computed is converted into Boolean circuit using AND, XOR gates. Then a garbler encrypts the circuit's gates into encrypted tables and creates encrypted versions of values. After garbling the key is distributed through oblivious transfer, so that it does not reveals their actual input to others or the garbler. Then the parties evaluate the circuit gate-by-gate, using their input keys and garbled tables, to produce an encrypted output key. Finally, the key is decrypted to get the result.

4. Homomorphic encryption: It allows multiple parties to jointly compute a function on their private data without revealing individual inputs by performing operations like addition or multiplication directly on encrypted data. It is commonly used for the hybrid approaches of secure multiparty computation techniques. Its working is based on the encryption as the initial step, that includes, each party encrypts their individual data using shared keys thereby applying homomorphic operations directly on these ciphertexts. The final encrypted result is shared and the parties collectively decrypt the final output not the individual inputs.

5. Zero-Knowledge Proofs (ZKP): It enables multiple parties to jointly compute a function on their private data, without revealing the underlying sensitive inputs or intermediate results.

6. Secure Function Evaluation (SFE): it allows parties to jointly compute a function over their inputs without revealing those inputs to each other.

7. Differential Privacy (DP):it combines noise in addition with secure protocols to allow analysis of shared private data while preserving individual privacy and security [5]. It prevents the concept of central server or colluding parties from learning sensitive details often by adding noise to inputs or gradients during the training. By using the SMPC to keep these noisy values hidden until the final aggregate is computed. Differential privacy ensures that the output of an analysis contains noise like gaussian noise etc. SMPC protocols allow multiple parties to jointly compute a function over their inputs without revealing the individual inputs.

II. LITERATURE REVIEW

1. This study describes about the protocols for secure multiparty computation and the interaction of parties to compute a joint function of their private inputs without revealing [6].

2. This paper describes about the network collaborative computing including techniques of SMPC like secret sharing, homomorphic cryptosystem, oblivious transfer etc. And discussed about the applications of SMPC in electronic voting, threshold signature, datamining, database queries [7].

3. This work provides a two round SMPC protocol that uses multilinear maps based on the learning from errors assumption (LWE). Here the protocol achieves sublinear communication overhead in polynomial time [8].

The above were the few works related to secure multiparty computation technique.

III. PROPOSED METHODOLOGY

The proposed methodology combines theoretical analysis, protocol evaluation, and experimental performance assessment. We had already covered the fundamental principles behind the secure multiparty computation, based on the techniques shared the proposed methodology are designed to evaluate SMC techniques under realistic scenarios, privacy-preserving data aggregation, and confidential financial computation.

Case study selection:

Privacy-preserving data aggregation (PPDA): It is a crucial application of SMPC that enables multiple data owners to collaboratively compute aggregate results over their private data without revealing individual inputs. In modern data driven environment, organizations frequently need to ana lyse distributed and sensitive data, such as medical documents, financial transactions, energy consumption and data, or behavior statistics. Traditional centralized aggregation methods contain serious privacy risk, as raw data must be shared with a trusted aggregator. PPDA eliminates this risk by ensuring data confidentiality throughout the computation process.

Secure multiparty computation provides cryptographic protocols that allow parties to jointly compute a function while keeping their inputs private. When applied to aggregation tasks such as computing sums, averages, histograms, or statistical metrics, SMC enables accurate results without exposing individual records.

In PPDA, each participant or party holds a private dataset D_i. the goal is to compute an aggregate function f (D_1, D_2, D_3, D_n), such as total sum, mean or variance, frequency counts, minimum or maximum counts without revealing any individual D_i [9]. Privacy is preserved using cryptographic techniques that ensures input privacy, output correctness, and resistance to adversarial attacks.

Secret sharing is one of the most widely used PPDA techniques, each data value is divided into multiple random shares and distributed among participants or computation servers [10]. For example, if a hospital wants to contribute the number of patients such as 120 as count, it splits this value into random shares such as 40, 20, 60 and sends them to different computation nodes. Aggregation is performed on shares, and final sum is reconstructed.

Another kind of PPDA technique is Homomorphic encryption-based aggregation, it allows computation to be performed directly on encrypted data. Aggregators can sum or multiply encrypted values without decrypting them []. For example, in a smart grid, households encrypt their electricity consumption data using a public key. The utility company aggregates encrypted values and decrypt only the final total consumption.

Consider a healthcare dataset example, here multiple hospital wants to compute the average blood sugar level of diabetic patients across all instituitions without sharing individual patient records or even hospital level raw statistics. Sharing such sensitive medical data would violate privacy and protection rules and regulations. Suppose there are three entities hospital A, B, C with 500,700,300 patients respectively and each hospital holds patient blood sugar level measurements locally.

By applying secure multiparty computation method,

Step 1: Local computation- computes the data locally without sharing patient-level data:

Partial sum (S_i)- sum of blood sugar levels of all diabetic patients,Patient count (N_i)- number of diabetic patients.

Table I. Local computation -Patient -level data

Hospital	Patient Count (N_i)	Partial Sum (S_i)
A	500	72,000
B	700	101,500
C	300	43,500

Step 2: Secret sharing: instead of sending (S_i) and (N_i) directly, each hospital applies a secret sharing scheme like Shamir's secret sharing. Here each value split into multiple random shares and are distributed among computation parties.

Step 3: secure average computation- find the average of blood sugar for the above data, that is, $\frac{\text{S total}}{\text{N total}}$.

Where S_{total} = 72000+101500+43500=217000

N_{total}= 500+700+300=1500

Average=144.67

This final average is revealed to all the parties.

IV. DESIGN AND IMPLEMENTATION METHOD

1. System architecture:

The SMPC framework is designed to allow multiple mutually distrustful parties to jointly compute a function over their private inputs without revealing sensitive data. The system follows a distributed architecture, where each participant locally holds its dataset and interacts with others only through the cryptographically protected messages. The primary goal includes privacy correctness, scalability, fault tolerance, and resistance against semi-honest adversaries. The design ensures that intermediate computation results remain protected throughout the protocol execution.

2. Threat model and assumptions:

The system operates under a semi-honest adversary model, where parties follow protocol specifications but may attempt to infer private data from observed messages. Secure communication channels and authenticated participants are assumed.

3. Protocol design overview:

The multiparty computation protocol is divided into four main parts, initialization and set up, input sharing, secure computation, result reconstruction. The corresponding pseudocode algorithms are explained in detail as below:

Algorithm 1: SystemInitialization- initializes all the global parameters.
Input: Number of parties n, threshold t

Output: Pubic parameters PP
1: Select a finite field F with large prime p
2: Define security parameter λ
3: Broadcast PP= {F, p, t, λ} to all parties
4: return PP

Algorithm 2: SecretShareInput-each party splits its private input into multiple shares using threshold secret sharing.
Input: Private input xi, threshold t, number of parties n
Output: Shares {si1, si2, sin}
1: Randomly select a polynomial f(x) of degree (t-1)
2: set f (0) = xi
3: for j=1 ton do
4: sij= f(j)
5: securely send sij to Party j
6: end for
7: return {si1, si2, sin}.

Algorithm 3: SecureAggregation - computes an aggregate function such as sum over shared inputs.
Input: Received shares {s1j, s2j, snj} at Party j
Output: Partial aggregated share Sj
1: Sj = 0
2: for I = 1 ton do
3: sj= sj + sij
4: end for
5: return Sj
Algorithm 4: ResultReconstruction- reconstruction of the final result.
Input: Partial results {S1, S2, S3, Sk}
Output: Final aggregate result R

1: collect at least t valid partial results
2: Apply polynomial interpolation
3: Recover R= f (θ)
4: return R
The protocol is implemented using a modular architecture separating cryptographic primitives from application logic. Each party executes local computation independently and participates in synchronized communication rounds. The system supports configurable parameters such as participant count, threshold values, and function types.

V. CONCLUSION

As an effective framework, this paper analyzes security and privacy aspects in secure multiparty computation (SMPC), emphasizing its role as a functional technology for privacy -preserving collaborative computation in distributed and untrusted environments. As data-driven decision making becomes increasingly across sensitive domains such as healthcare etc., the need to compute over private data without revealing confidentiality has become critical. SMPC addresses this challenge by enabling multiple parties to jointly evaluate a function while ensuring that no participant reveals individual data. Through this analysis SMPC provides strong theoretical guarantees for data privacy, making it an alternative tool to centralized or distributed data aggregation models. From a privacy perspective, the study demonstrated that cryptographic techniques such as secret sharing, homomorphic encryption plays a central role in preventing information leakage during computation.

REFERENCES

[1] [1] Xiaoqiang Guo, Shuai Zhang, Ying Li- Key Technologies and Applications of Secure Multiparty Computation, ijeecs, Vol. 11, No. 7, July 2023. pp. 3774-3779.

[2] [2] Hiroki Kaminaga et.al -MPCFL: Towards Multi-party Computation for Secure Federated Learning Aggregation, UCC '23: Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing (December 2023).

[3] [3] Jimin Kang, -Secure Multi-Party Computation: Oblivious Transfer [online]. Available: https://medium.com/@jimin.kang821/secure-multi-party-computation-oblivious-transfer-3e819878ca1e

[4] Aner Ben-Efraim. Et.al., - MYao: Multiparty ``Yao" Garbled Circuits with Row Reduction, Half Gates, and Efficient Online Computation, [online], Available: https://eprint.iacr.org/2024/1430

[5] Chao Zheng,Et.al,-Optimizing Privacy in Federated Learning with MPC and Differential Privacy, CACML '24: Proceedings of the 2024 3rd Asia Conference on Algorithms,

Computing and Machine Learning,Pages 165 – 169.

[6] Yehuda Lindell, Secure Multiparty Computation (MPC). [online]Available: https://eprint.iacr.org/2020/300.pdf?utm_source=chatgpt.com.

[7] Xiaoqiang Guo, Shuai Zhang, Ying Li, Key Technologies and Applications of Secure Multiparty Computation, Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), p-ISSN: 2502-4752, e-ISSN: 2502-4760.

[8] Yun Luo, Yuling Chen, Tao Li, Chaoyue Tan & Hui Dou, (2024)- Cloud-SMPC: two-round multilinear maps secure multiparty computation based on LWE assumption, Journal of Cloud Computing: Advances, Systems and Applications, https://doi.org/10.1186/s13677-023-00586-5.

[9] Jing Wang, Libing Wu, Sherali Zeadally, Muhammad Khurram Khan, and Debiao He. 2021. Privacy-preserving Data Aggregation against Malicious Data Mining Attack for IoT-enabled Smart Grid. ACM Trans. Sen. Netw. 17, 3, Article 25 (June 2021), 25 pages. https://doi.org/10.1145/3440249

[10] Shalabi, E.; Khedr, W.;Rushdy E.; Salah, A. A Comparative Study of Privacy-Preserving Techniques in Federated Learning: A Performance and Security Analysis. Information 2025, 16, 244. https://doi.org/10.3390/info16030244