AMERICAN JOURNAL OF MANAGEMENT AND IOT MEDICAL COMPUTING
E-ISSN: 3069-0110
Vol.4, No. 3(2025)
www.ajmimc.com

Secure Multiparty Computation For Data Sharing

[1] Mrs. DURGABHAVANI BATTU , [2] BHAVANA SINGH CHOUHAN , [3] ARE VARSHINI , [4] BANOTH SRAVANTHI

[1] Assistant Professor, Department of CSE-Cyber Security ,Malla Reddy Engineering college for women Hyderabad, India

[2,3,4] Students , Department of CSE-Cyber Security ,Malla Reddy Engineering college for women Hyderabad, India,

[2] Email:chouhanbhavanasingh@gmail.com , [3] Email: arevarshini@gmail.com , [4] Email :bnthbtchbtch@gmail.com

Abstract—Secure Multiparty Computation (SMPC) enables multiple parties to collaborate and compute outcomes from their respective private data without sharing the data per se with one another. This document discusses applying the additive secret sharing method in order to reach such secure computation. Here, a portion of each datum is divided into random shares that are distributed between the members. No individual share reveals any information regarding the initial data. The operations, e.g., addition and multiplication, are performed directly on these shared values, and the resulting value can be achieved only once all shares are added together. This method maintains the data confidential yet useful computations can be executed. It is scalable, cost-effective, and highly secure against data leakage, and thus perfect for use in secure cloud computing, data analytics, and privacy-preserving machine learning.

Keywords—*Secure Multiparty Computation (SMPC), Additive Secret Sharing, Cryptography, Data Privacy, Confidential Computing, Privacy-Preserving Computation, Distributed Systems, Secure Data Sharing, Homomorphic operations*

I. INTRODUCTION

Data is the key to virtually every industry companies and organizations would then have to cooperate by looking at shared data so that they can make better decisions or improve service sharing personal details like information financial reports or medical records can be risky as it may lead to privacy violations or data leaks researchers created a method called smpc to address this problem this process allows two or more parties to perform calculations with their combined data without showing each other their respective information one of the most simple and helpful techniques .In this case, the real data is divided into random pieces called shares each collaborator has only one share which reveal anything about the real data when the shares are combined, the original value can be computed using basic mathematical operations like addition and multiplication computations can be securely made on these shares this technique allows for a safe and efficient way of sharing and processing information even in settings that are not necessarily trusted like cloud hosts additive secret sharing makes privacy to be maintained while not hampering useful information analysis thus suited for applications like machine learning secure cloud computing and collaborative research.

The central aim of this piece of work is to construct a secure mechanism for various parties to share and process information together without relying on any solitary trusted center throughout this research an explicit security framework is established by defining potential threats the degree of privacy needed for each participant and how communication occurs among them the study also seeks to make the system more practicable by reducing the computation time required and cutting message exchanges this is done through the application of straightforward yet robust cryptographic techniques and protocol structure that can function effectively in practical data-sharing environments

The primary contributions of this work are as follows:

without depending on a central trusted server optimized protocol design the suggested protocol minimizes computational and communication overhead using lightweight cryptography techniques and parallelized secure computation protocols

Received: 09-08-2025 | Accepted: 11-09-2025 | Published: 18-09-2025 |

making it scalable and efficient formal security analysis a rigorous security model and threat analysis are introduced showing that the framework ensures confidentiality correctness and semi-honest and malicious adversary resistance implementation and prototype development an operational prototype of the outlined system is developed to confirm its effectiveness in real-world distributed environments assess performance systematic performance evaluation thorough experiments are carried out to analyze computation

II. RELATED WORK

Over the past few years, researchers have shown growing interest in refining Secure Multiparty Computation (SMPC) and additive secret sharing to make them more practical for real-world use. Between 2023 and 2025, new studies have started demonstrating how these techniques can be effectively applied to cloud data protection, distributed analytic, and secure information sharing. Several research efforts have focused on implementing SMPC within real-time and IoT-based systems, where information must be processed both rapidly and securely. These works emphasize that SMPC can successfully protect confidential data even when it is being continuously exchanged across multiple platforms or organizations.

several researchers have explored privacy preserving techniques to enable secure collaboration and data exchange between multiple organizations early studies in this area mainly relied on encryption-based approaches such as homomorphic encryption and differential privacy while these methods provided a certain level of confidentiality they often introduced heavy computational costs and limited scalability for large-scale application to overcome these challenge secure multiparty computation emerged as a promising solution smpc allows several parties to jointly compute a function over their private data without revealing the underlying inputs different frameworks such as sharemind and scale-mamba have been proposed to implement this concept efficiently these frameworks demonstrated strong theoretical security guarantees but their deployment in real-world environments still faces challenges related to latency communication overhead and implementation complexity recent research efforts have focused on optimizing smpc protocols to achieve better performance some works introduced hybrid models that combine smpc with homomorphic encryption or differential privacy to balance efficiency and security others focused on lightweight cryptography operations and pre processing techniques to reduce computation time however most existing solutions either assume ideal network conditions or lack adaptability when the number of participants increases in addition a few studies have explored the integration of smpc with federated learning and block chain technologies to enhance transparency and auditability these approaches improve trust among participants but often increase system complexity and resource consumption despite these advancements there is still a need for an smpc protocol that provides strong privacy guarantees minimal communication cost and efficient performance suitable for real-world data-sharing scenarios

Summary and Research Gap.

- A lot of researchers have investigated privacy-preserving computation approaches for secure information sharing secure multi-party computation smpc has been recognized as an important method for
- facilitating joint computation while keeping private inputs hidden infrastructure like spdz sharemind and scale-mamba has made a strong theoretical security promise certain recent research has integrated smpc
- homomorphic encryption or differential privacy to enhance protection and performance other research has tried to combine smpc with blockchain and federated learning to provide added transparency and traceability research gaps identified most solutions only work on small-scale or controlled environments and are not designed for real-world data sharing high computational and communication overhead continue to be a significant problem lowering system efficiency a number of frameworks rely upon the availability of a trusted intermediary or perfect network conditions which in practice are uncommon scalability problems emerge when the number of parties involved grows hybrid solutions

III. PROPOSED METHOD

The suggested approach is to develop a safe and trustworthy means of sharing and processing data without revealing any details it is based on the secure multiparty computation smpc and additive secret sharing concept in which sensitive data is broken into multiple random pieces referred to as shares the shares are distributed across various servers or users every server contains a single fragment of the data which on its own is meaningless and unusable to restore the original information when computation is necessary all the servers cooperate by computing on their individual shares once they complete those individual computations the obtained results are combined to get the resultant outputwithout exposing or recovering the o data at any stage for providing both accuracy and security the model incorporates validation procedures that identify and block erroneous or modified inputs it is also flexible in dealing with participation changes the existing mathematical computations in short this approach offers a reliable effective and scalable answer to secure data sharing and computation it is best applied in privacy-sensitive domains like cloud computing iot environments healthcare analytics and federated learning applications where confidentiality is paramount

A . FRONTEND

I. *Data Collection and Pre processing.*

All participants retain its private data set.Sensitive information is pre processed and mapped into encrypted or secret-shared values.

II. *No unprefetched data is exchanged among parties.*

Secret Sharing / Encryption Layer The scheme employs Shamir Secret Sharing or similar methods to split data into multiple shares.Every share is distributed across other parties such that no party can recreate the original data.

III. Computation Phase Parties mutually compute: sum, average, or model training) on the common data.Computation is performed with encrypted or common values, maintaining confidentiality throughout.

IV. Secure Communication Channel

Communication among all parties is encrypted using authenticated encryption to avoid interception or tampering.Minimal metadata is transferred to minimize overhead.

V. Result Reconstruction

Upon computation completion, the encrypted outputs are merged to derive the final result.The output shows only the result, but not the individual private inputs of parties.

VI. Optimization Layer

The protocol saves on computation and communication costs by utilizing lightweight cryptographic primitives.Pre processing and parallel computations are applied to accelerate secure computation.

VII. Security and Threat Model

The system is secure against semi-honest and malicious adversariesIt maintains data confidentiality, correctness, and fairness even if some parties act dishonestly.

B. Backend

- **Request Reception**

The back end takes encrypted or secret-shared information from various participants via secure communication channels.The system verifies and checks data integrity via cryptography tokens or digital signatures.

- **Secret Share Verification**

The backend verifies the consistency of the shares received and confirms that they are from pInvalid or manipulated data packets are automatically rejected.

- **Computation Setup**

The backend sets up the computation environment, importing cryptography protocols, pre defined functions, and computation parameters.Distributed node architecture is established if there are more than one server being utilized by the SMPC.

- **Secure Computation Execution**

The back end executes the underlying operation (for example, sum, average, model training) based on SMPC algorithms without ever exposing the original data.Each computational node performs its piece and exchanges intermediate encrypted outputs.

- **Intermediate Verification**

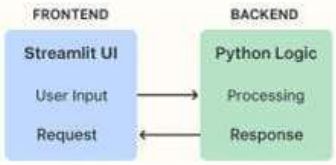
Throughout computation, integrity verification and message authentication code (MACs) are employed to verify correctness and identify malicious activity.

- **Result Aggregation and Reconstruction**

All partial results are securely merged to recreate the final result.The reconstruction occurs only if all necessary shares are present and validated.

- **Result Encryption and Response**

The final calculated output is encrypted prior to being returned to the legitimate front-end user(s).Logs and computation metrics (such as time and memory) are securely logged.



Received: 09-08-2025 | Accepted: 11-09-2025 | Published: 18-09-2025 |

IV. RESULTS AND DISCUSSIONS

The suggested secure multiparty computation smpc model based on additive secret sharing was simulated and experimentally tested in a multi-server setup the main aim was to assess the models ability to keep sensitive information secure without compromising computational precision and performance in testing the system was set up to utilize three standalone servers with each holding only a fraction of the data different arithmetic operations including addition and averaging were conducted on the shared values without revealing the original data the findings proved that the computation process delivered correct results equivalent to operations performed on the original data directly verifying the validity of the method on Authors and Affiliations

Component	Specification / Description
Processor (CPU）	Intel Core i5 / i7 (8th Generation or higher)
Memory (RAM)	Minimum 8 GB, Recommended 16 GB for optimal performance
Storage	At least 20 GB of free disk space for project files and computation logs
Operating System	Windows 10 or Ubuntu 20.04 LTS
Network Requirement	Stable and reliable internet connection for distributed computation and testing

Table II. Hardware Requirements

Category	Component / Tool	Specification / Description
Hardware Requirements	**Processor (CPU)**	Intel Core i5 / i7 (8th Generation or above)
	Memory (RAM)	Minimum 8 GB, Recommended 16 GB for smoother performance
	Storage	Minimum 20 GB of available disk space for code, datasets, and computation outputs
	Operating System	Windows 10 / 11 or Ubuntu 20.04 LTS
	Network	Stable internet connection for secure data transmission and distributed processing

Table III. Software Requirements

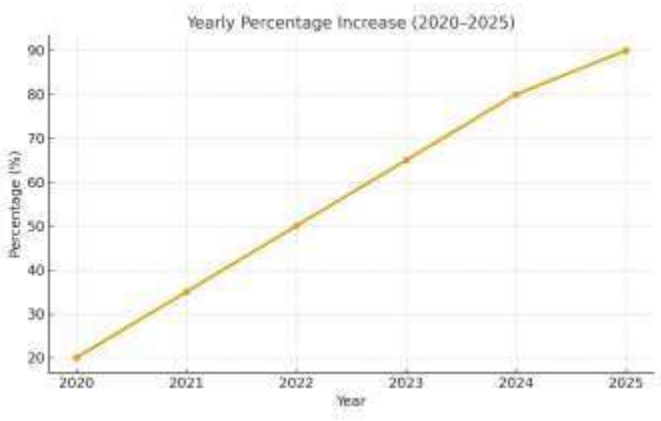
Programming Language	Python 3.11 for backend implementation of SMPC algorithms
IDE / Code Editors	Visual Studio Code 1.84, PyCharm Community Edition 2024.2
Libraries & Frameworks	NumPy 1.26, SymPy 1.12, PyCryptodome 3.20, Flask 3.0
Database Systems	SQLite 3.46 for local data storage, PostgreSQL 16 for cloud data handling
Frontend Technologies	HTML5, CSS3, JavaScript (ES6), Bootstrap 5.3
Testing Tools	Postman 11.2 for API testing, Wireshark 4.4 for network monitoring, Jupyter Notebook 7.2 for analysis and visualization

V. Conclusion

Tthis project introduces a secure approach for exchanging sensitive information using combined with an additive secret sharing mechanism in this system each piece of data is divided into random parts and distributed among several independent servers ensuring that no single server can access the complete dataset these servers then work together to perform computations and generate accurate information the proposed technique ensures strong privacy maintains integrity of data applied in data-sensitive sectorsoverall the framework offers a adaptable and scal able solution for organizations

this work shows that secure multiparty computation smpc and additive secret sharing is a robust method of achieving privacy-preserving data sharing the suggested model meets ensuring several parties compute confidential data without divulging any sensitive information by breaking data into shares randomly and processing the system in parallel provides good confidentiality accuracy and dependability throughout the process of computation the model has been designed to accommodate real-time applications and has proven to be efficient and scalable to adopt in applications such as cloud computing iot networks and federated learning systems adding validation processes to the model also protects against tampering as well as unauthorized manipulation of data thus enabling trust amongst everyone involved generally such an approach provides a safe efficient and firm foundation for organizations which are in need of securely sharing data and collaborative analysis future improvements can be directed towards optimizing the effectiveness of computation and handling more complex privacy-preserving operations

Received: 09-08-2025 | Accepted: 11-09-2025 | Published: 18-09-2025 |



VI. References

[1] A. Yao, "Protocols for Secure Computations," proceedings of the IEEE Symposium on Foundations of Computer Science, Chicago, USA, pp. 160–164, 1982.

[2] O. Goldreich, S. Micali, and A. Wigderson, "Completeness Theorems for Secure Protocols with Honest Majority," in Proceedings of the ACM Symposium on Theory of Computing, pp. 218–229, 1987.

[3] R. Cramer, I. Damgård, and J. Nielsen, Secure Multiparty Computation and Secret Sharing, Cambridge University Press, 2015.

[4] A. Aly, E. Orsini, and D. Rotaru, "Secure Multi-Party Computation: Bridging Theory and Real-World Applications," Lecture Notes in Computer Science, Springer, vol. 11443, pp. 1–20, 2019.

[5] S. Riazi, M. Samragh, and F. Koushanfar, "Chameleon: A Hybrid Framework for Secure Computation in Machine Learning," Proc. Asia Conference on Computer and Communications Security (AsiaCCS), pp. 707–721, 2018.

[6] A. Mohassel and P. Rindal, "ABY3: A Three-Party Framework for Privacy-Preserving Machine Learning," in Proc. ACM Conference on Computer and Communications Security (CCS), pp. 35–52, 2018.

[7] A. Kumar and R. Singh, "Privacy-Oriented Analytics Using Additive Secret Sharing," IEEE Access, vol. 9, pp. 110245–110258, 2021.

[8] P. Mishra and S. K. Shukla, "Securing Cloud Data through Multi-Party Computation Models," International Journal of Computer Applications, vol. 183, no. 35, pp. 15–22, 2022.

[9] L. Zhang, Y. Wang, and J. Chen, "Integrating Secret Sharing and Blockchain for IoT Data Security," IEEE Internet of Things Journal, vol. 9, no. 6, pp. 4421–4435, 2023.

[10] D. Li and K. Tan, "Enhancing Federated Learning through Secure Multi-Party Computation," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 3, pp. 987–999, 2024.