Advances in Computer Sciences
Vol. 7 (2024)
https://academicpinnacle.com/index.php/acs

A Comparative Assessment of Multi-Party Computation Protocols: Performance Metrics and Security Implications

William K. Kwaku
Department of Computer Science, University of Ghana, Ghana

Abstract

Multi-Party Computation (MPC) protocols enable secure computations across multiple parties while preserving the privacy of their inputs. This paper provides a comparative assessment of various MPC protocols by evaluating their performance metrics and security implications. The study focuses on established protocols such as Yao's Garbled Circuits, Secure Multi-Party Computation (SMPC) with Homomorphic Encryption, and Secret Sharing Schemes. We analyze their efficiency in terms of computational and communication overhead, as well as their robustness against different types of attacks. The findings aim to guide the selection of appropriate MPC protocols for various applications based on specific security requirements and performance constraints.

Keywords: Multi-Party Computation (MPC), Secret Sharing, Homomorphic Encryption, Zero-Knowledge Proofs (ZKP), Performance Metrics, Computational Overhead, Communication Overhead, Security Guarantees, Security Models.

1. Introduction:

Multi-Party Computation (MPC) has emerged as a pivotal technology in the realm of secure computation, enabling multiple parties to collaboratively compute a function over their combined inputs while preserving the privacy of each participant's data[1]. As data privacy concerns intensify and the demand for secure collaborative processes grows, MPC provides a robust framework for ensuring that sensitive information remains confidential even during joint computations. The significance of MPC extends across various domains, including secure financial transactions, privacy-preserving data analysis, and collaborative machine learning. Despite its importance, selecting the

appropriate MPC protocol for a given application involves balancing complex trade-offs between performance and security.

The primary objective of this paper is to conduct a comprehensive comparative assessment of existing MPC protocols, with a focus on evaluating their performance metrics and security implications[2, 3]. By analyzing protocols such as Yao's Garbled Circuits, Secure Multi-Party Computation with Homomorphic Encryption, and Secret Sharing Schemes, we aim to provide insights into their respective strengths and weaknesses. This evaluation is crucial for understanding how different protocols perform under varying computational and communication constraints, and how they withstand diverse security threats. The results of this study will guide researchers and practitioners in choosing the most suitable MPC protocol for their specific needs, based on detailed considerations of efficiency and security.

The structure of this paper is organized as follows: we begin with an overview of key MPC protocols and their underlying mechanisms[4]. Next, we delve into an analysis of their performance metrics, including computational and communication overhead, and scalability. Following this, we explore the security implications of each protocol, focusing on threat models, robustness against attacks, and privacy guarantee. Finally, we present a comparative analysis of the protocols, discuss the trade-offs between performance and security, and offer recommendations based on various application scenarios. Through this detailed examination, we aim to contribute valuable insights to the ongoing development and deployment of secure multi-party computation systems[5].

2. MPC Protocols Overview:

Homomorphic encryption-based protocols leverage advanced cryptographic techniques to perform computations on encrypted data without the need for decryption, thus preserving the privacy of the data throughout the process[6]. A key example is Paillier encryption, which supports additive homomorphism, allowing operations such as secure summation of encrypted values[7]. This scheme enables parties to compute aggregates over encrypted inputs while maintaining data confidentiality. Another significant approach is the BGV scheme (Brakerski-Gentry-Vaikuntanathan), which provides fully homomorphic encryption (FHE) capabilities, supporting both additive and multiplicative operations on encrypted data[8]. This versatility allows for more complex computations compared to additive homomorphic encryption.

Homomorphic encryption-based protocols are highly valued for their ability to ensure strong data privacy, making them suitable for sensitive applications such as cloud computing and secure data analysis. However, they often incur substantial computational overhead and increased communication costs due to the complexity of the encryption and decryption processes[9, 10]. These performance challenges, while mitigated by ongoing research and optimizations, remain a critical consideration when deploying homomorphic encryption in practical scenarios.

Protocols utilizing zero-knowledge proofs (ZKPs) represent a sophisticated method for achieving secure computations by allowing parties to prove the validity of certain statements without revealing any additional information. Zero-knowledge proofs are cryptographic techniques that enable a prover to convince a verifier that a statement is true without disclosing the underlying data or specifics of the proof. In the context of multi-party computation, ZKP-based protocols ensure that the computations are executed correctly while preserving the privacy of the inputs and intermediate results[11, 12]. Notable examples include protocols based on zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) and zk-STARKs (Zero-Knowledge Scalable Transparent Arguments of Knowledge). zk-SNARKs offer succinct proofs with short verification times, making them efficient for scenarios requiring quick validation, whereas zk-STARKs provide transparency and scalability by eliminating the need for a trusted setup and allowing for more robust security guarantees. These protocols are particularly useful in applications such as blockchain and privacy-preserving smart contracts, where both security and efficiency are paramount[13, 14]. Despite their advantages, ZKP-based protocols can be computationally intensive and require significant cryptographic expertise to implement effectively. Ongoing research aims to address these challenges by improving performance and reducing the computational burden associated with zero-knowledge proofs.

3. Performance Metrics:

Computational overhead refers to the additional computational resources required by a protocol to perform its operations compared to a non-secure or baseline approach[15]. In the context of multi-party computation (MPC) protocols, computational overhead encompasses the extra processing time and resource consumption needed for cryptographic operations, such as encryption, decryption, and secure function evaluations. For instance, protocols based on homomorphic encryption often involve complex

mathematical operations on encrypted data, which can be significantly more resource-intensive than operations on plaintext data. Similarly, secret sharing schemes may require extensive computational effort for generating and managing shares, as well as for reconstructing the secret[16]. Zero-knowledge proof protocols, while providing strong privacy guarantees, can also introduce considerable computational overhead due to the complexity of the proofs and the necessity for extensive cryptographic processing[17]. Efficiently managing computational overhead is crucial for ensuring that MPC protocols remain practical and scalable, especially in scenarios involving large datasets or numerous participants[18]. Advances in cryptographic techniques and optimization strategies continue to seek ways to minimize this overhead while maintaining robust security guarantees.

Communication overhead refers to the additional network resources required for exchanging data among participants in a multi-party computation (MPC) protocol[19]. This includes the amount of data transmitted between parties, as well as the frequency and size of communication rounds necessary to complete the computation. In MPC protocols, communication overhead can be substantial, especially for protocols involving secret sharing or homomorphic encryption. For instance, secret sharing-based protocols may require each participant to send and receive multiple shares of the secret, resulting in increased data traffic[20, 21]. Similarly, homomorphic encryption protocols often involve transmitting encrypted data and intermediate results, which can lead to significant bandwidth consumption. Zero-knowledge proof protocols also contribute to communication overhead due to the need for exchanging proof data and verification messages[22]. High communication overhead can impact the efficiency and scalability of MPC protocols, particularly in environments with limited bandwidth or when scaling up to large numbers of participants. Addressing communication overhead through optimized protocol design and efficient data handling techniques is essential to improving the practicality and performance of secure multi-party computations[23].

4. Security Implications:

Security guarantees in multi-party computation (MPC) protocols ensure that the privacy and integrity of participants' data are maintained throughout the computation process. These guarantees typically include confidentiality, integrity, and robustness. Confidentiality ensures that the inputs and intermediate results of the computation remain hidden from all parties except those explicitly authorized to see them[24]. This is achieved through techniques such as encryption and secret sharing, which prevent unauthorized access to

sensitive information. Integrity guarantees that the computation is performed correctly and that the results are accurate, even if some parties attempt to disrupt or corrupt the process. Robustness refers to the protocol's ability to handle adversarial behavior, including malicious participants who may try to breach security or manipulate the outcomes. Different security models, such as the semi-honest model and the malicious model, provide varying levels of assurance[25]. The semi-honest model assumes that parties follow the protocol but may attempt to learn additional information, while the malicious model accounts for participants who actively try to subvert the protocol. Ensuring strong security guarantees is crucial for the effective deployment of MPC protocols in sensitive applications, balancing the need for privacy with the requirement for reliable and accurate results[26].

Security models in multi-party computation (MPC) define the assumptions and guarantees regarding the behavior of participants and the protection of data. The two primary models are the semi-honest model and the malicious model[27]. The semi-honest model, also known as the honest-but-curious model, assumes that participants follow the protocol correctly but may attempt to infer additional information from the data they receive. This model provides security guarantees under the assumption that participants are honest in their execution of the protocol but curious about the private inputs and outputs of others. In contrast, the malicious model accommodates participants who may deliberately deviate from the protocol or attempt to disrupt the computation for personal gain[28]. This model requires more robust mechanisms to ensure that the protocol can withstand malicious attacks and still produce correct and reliable results. Security models influence the design and complexity of MPC protocols, with the malicious model generally requiring more sophisticated techniques and protocols to provide the same level of security as the semi-honest model. Understanding these models is crucial for selecting appropriate MPC protocols based on the security requirements and the trust level among participants in various applications[29].

5. Comparative Analysis:

Performance comparison in multi-party computation (MPC) involves evaluating different protocols based on their efficiency in terms of execution time, computational resources, and communication overhead. Execution time refers to the total duration required to complete the computation, which can be influenced by factors such as the complexity of cryptographic operations and the protocol's inherent design[30]. Computational resources include the CPU and memory usage during the computation, which varies significantly among

protocols depending on their underlying cryptographic techniques. For example, protocols based on homomorphic encryption often incur high computational costs due to complex operations on encrypted data, while secret sharing-based protocols may involve less intensive computations but require managing multiple shares[31]. Communication overhead, which includes the amount of data exchanged and the number of communication rounds, also plays a critical role in performance. Protocols with high communication overhead may be less practical in environments with limited bandwidth or when scaling to large numbers of participants. A thorough performance comparison helps in selecting the most suitable MPC protocol for a given application, balancing efficiency with the necessary security guarantees to meet specific requirements[32].

Security comparison in multi-party computation (MPC) focuses on assessing the robustness of different protocols against various types of attacks and their ability to protect participants' data throughout the computation process. This comparison involves evaluating how well each protocol upholds confidentiality, integrity, and robustness under different security models. Confidentiality measures how effectively a protocol prevents unauthorized access to sensitive data, ensuring that inputs and intermediate results remain private. Integrity guarantees that the computation is executed correctly and the results are accurate, even if some participants act maliciously[33]. Robustness assesses the protocol's ability to handle adversarial behavior, including participants who may attempt to disrupt or manipulate the computation. Protocols operating under the semi-honest model are designed to handle honest-but-curious participants, whereas those under the malicious model are equipped to counteract deliberate attempts to compromise the protocol. By comparing these aspects, one can determine which protocol offers the most appropriate security guarantees for different applications, considering factors such as the trust level among participants and the sensitivity of the data being processed[34]. This comparison is essential for selecting the right MPC protocol to ensure both security and practical feasibility.

6. Conclusion:

In conclusion, the comparative assessment of multi-party computation (MPC) protocols reveals significant insights into their performance and security trade-offs, essential for selecting the most appropriate solution for various applications. Secret sharing-based protocols offer robust confidentiality through the distribution of data shares but may involve substantial communication overhead. Homomorphic encryption-based protocols provide

strong privacy by allowing computations on encrypted data, though they often come with high computational costs. Zero-knowledge proof protocols ensure rigorous security guarantees by allowing parties to verify computations without revealing sensitive information, yet they can be complex and resource-intensive. The choice of MPC protocol depends on balancing these factors—performance metrics such as execution time, computational and communication overhead, and security guarantees including confidentiality, integrity, and robustness. As the field of MPC continues to evolve, ongoing research and advancements are likely to enhance the efficiency and security of these protocols, making them increasingly practical for a wide range of applications. Understanding these dynamics is crucial for implementing secure and efficient multi-party computations in sensitive and data-intensive environments.

References:

[1] S. Dodda, N. Kunchakuri, A. Kumar, and S. R. Mallreddy, "Automated Text Recognition and Segmentation for Historic Map Vectorization: A Mask R-CNN and UNet Approach," *Journal of Electrical Systems,* vol. 20, no. 7s, pp. 635-649, 2024.

[2] A. Kumar, S. Dodda, N. Kamuni, and V. S. M. Vuppalapati, "The Emotional Impact of Game Duration: A Framework for Understanding Player Emotions in Extended Gameplay Sessions," *arXiv preprint arXiv:2404.00526,* 2024.

[3] Y. Alexeev *et al.*, "Quantum computer systems for scientific discovery," *PRX quantum,* vol. 2, no. 1, p. 017001, 2021.

[4] A. A. Mir, "Transparency in AI Supply Chains: Addressing Ethical Dilemmas in Data Collection and Usage," *MZ Journal of Artificial Intelligence,* vol. 1, no. 2, 2024.

[5] S. Lad, "Cybersecurity Trends: Integrating AI to Combat Emerging Threats in the Cloud Era," *Integrated Journal of Science and Technology,* vol. 1, no. 8, 2024.

[6] S. Dodda, A. Kumar, N. Kamuni, and M. M. T. Ayyalasomayajula, "Exploring Strategies for Privacy-Preserving Machine Learning in Distributed Environments," *Authorea Preprints,* 2024.

[7] D. Bogdanov, M. Niitsoo, T. Toft, and J. Willemson, "High-performance secure multi-party computation for data mining applications," *International Journal of Information Security,* vol. 11, pp. 403-418, 2012.

[8] K. Peng *et al.*, "Towards making the most of chatgpt for machine translation," *arXiv preprint arXiv:2303.13780,* 2023.

[9] H. Shah and N. Kamuni, "DesignSystemsJS-Building a Design Systems API for aiding standardization and AI integration," in *2023 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA)*, 2023: IEEE, pp. 83-89.

[10] A. A. Mir, "Sentiment Analysis of Social Media during Coronavirus and Its Correlation with Indian Stock Market Movements," *Integrated Journal of Science and Technology,* vol. 1, no. 8, 2024.

[11] J. S. Arlagadda Narasimharaju, "SystemC TLM2. 0 modeling of network-on-chip architecture," Arizona State University, 2012.

[12] L. Braun, D. Demmler, T. Schneider, and O. Tkachenko, "Motion–a framework for mixed-protocol multi-party computation," *ACM Transactions on Privacy and Security,* vol. 25, no. 2, pp. 1-35, 2022.

[13] S. Dodda, A. Kumar, N. Kamuni, and M. M. T. Ayyalasomayajula, "Exploring Strategies for Privacy-Preserving Machine Learning in Distributed Environments."

[14] Q. Zhong, L. Ding, J. Liu, B. Du, and D. Tao, "Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert," *arXiv preprint arXiv:2302.10198,* 2023.

[15] A. Kumar, S. Dodda, N. Kamuni, and R. K. Arora, "Unveiling the Impact of Macroeconomic Policies: A Double Machine Learning Approach to Analyzing Interest Rate Effects on Financial Markets," *arXiv preprint arXiv:2404.07225,* 2024.

[16] A. A. Mir, "Optimizing Mobile Cloud Computing Architectures for Real-Time Big Data Analytics in Healthcare Applications: Enhancing Patient Outcomes through Scalable and Efficient Processing Models," *Integrated Journal of Science and Technology,* vol. 1, no. 7, 2024.

[17] S. S. Gill *et al.*, "AI for next generation computing: Emerging trends and future directions," *Internet of Things,* vol. 19, p. 100514, 2022.

[18] N. Kamuni, I. Cruz, Y. Jaipalreddy, R. Kumar, and V. Pandey, "Fuzzy Intrusion Detection Method and Zero-Knowledge Authentication for Internet of Things Networks," *International Journal of Intelligent Systems and Applications in Engineering,* vol. 12, no. 16s, pp. 289-296, 2024.

[19] S. Lad, "Harnessing Machine Learning for Advanced Threat Detection in Cybersecurity," *Innovative Computer Sciences Journal,* vol. 10, no. 1, 2024.

[20] A. A. Mir, "Adaptive Fraud Detection Systems: Real-Time Learning from Credit Card Transaction Data," *Advances in Computer Sciences,* vol. 7, no. 1, 2024.

[21] A. Katal, S. Dahiya, and T. Choudhury, "Energy efficiency in cloud computing data centers: a survey on software technologies," *Cluster Computing,* vol. 26, no. 3, pp. 1845-1875, 2023.

[22] S. Bhattacharya, S. Dodda, A. Khanna, S. Panyam, A. Balakrishnan, and M. Jindal, "Generative AI Security: Protecting Users from Impersonation and Privacy Breaches," *International Journal of Computer Trends and Technology,* vol. 72, no. 4, pp. 51-57, 2024.

[23] S. Dahiya, "Cloud Security Essentials for Java Developers Protecting Data and Applications in a Connected World," *Advances in Computer Sciences,* vol. 7, no. 1, 2024.

[24] N. Kamuni, S. Dodda, S. Chintala, and N. Kunchakuri, "Advancing Underwater Communication: ANN-Based Equalizers for Improved Bit Error Rates," *Available at SSRN 4886833,* 2022.

[25] S. Dahiya, "Developing AI-Powered Java Applications in the Cloud Harnessing Machine Learning for Innovative Solutions," *Innovative Computer Sciences Journal,* vol. 10, no. 1, 2024.

[26] N. Kamuni, M. Jindal, A. Soni, S. R. Mallreddy, and S. C. Macha, "Exploring Jukebox: A Novel Audio Representation for Music Genre Identification in MIR," in *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)*, 2024: IEEE, pp. 1-6.

[27] J. S. Arlagadda and N. Kamuni, "Hardware-Software Co-Design for Efficient Deep Learning Acceleration," *MZ Computing Journal,* vol. 4, no. 1, 2023.

[28] N. Kamuni and J. S. Arlagadda, "Exploring Multi-Agent Reinforcement Learning: Techniques, Applications, and Future Directions," *Advances in Computer Sciences,* vol. 4, no. 1, 2021.

[29] S. Dahiya, "Java in the Cloud: Best Practices and Strategies Optimizing Code for Performance and Scalability," *MZ Computing Journal,* vol. 5, no. 2, 2024.

[30] N. Kamuni and D. Panwar, "Enhancing Music Genre Classification through Multi-Algorithm Analysis and User-Friendly Visualization," *arXiv preprint arXiv:2405.17413,* 2024.

[31] J. S. Arlagadda and N. Kamuni, "Harnessing Machine Learning in Robo-Advisors: Enhancing Investment Strategies and Risk Management," *Journal of Innovative Technologies,* vol. 5, no. 1, 2022.

[32] S. Dahiya, "Safe and Robust Reinforcement Learning: Strategies and Applications," *Innovative Computer Sciences Journal,* vol. 9, no. 1, 2023.

[33] A. Ucar, M. Karakose, and N. Kırımça, "Artificial intelligence for predictive maintenance applications: key components, trustworthiness, and future trends," *Applied Sciences,* vol. 14, no. 2, p. 898, 2024.

[34] J. S. A. Narasimharaju, "Smart Semiconductor Wafer Inspection Systems: Integrating AI for Increased Efficiency."