SPAD – A Secure and Privacy-Preserving Distributed Analytics Framework

1st Imran Makhdoom
SEDE, FEIT
University of Technology Sydney
Ultimo, NSW, 2007, Australia
imran.makhdoom@uts.edu.au

2nd Mehran Abolhasan
SEDE, FEIT
University of Technology Sydney
Ultimo, NSW, 2007, Australia
Mehran.Abolhasan@uts.edu.au

3rd Justin Lipman
SEDE, FEIT
University of Technology Sydney
Ultimo, NSW, 2007, Australia
Justin.Lipman@uts.edu.au

4th Daniel Franklin
SEDE, FEIT
University of Technology Sydney
Ultimo, NSW, 2007, Australia
Daniel.Franklin@uts.edu.au

5th Massimo Piccardi
SEDE, FEIT
University of Technology Sydney
Ultimo, NSW, 2007, Australia
Massimo.Piccardi@uts.edu.au

***Abstract*—Conventional secure multi-party computation schemes depend on trusted parties and have high complexity, transparency, and data integrity concerns. This research presents SPAD, a secure, distributed, and privacy-preserving data storage and analytics framework designed to address the above challenges. SPAD leverages Shamir's secret-sharing scheme, Paillier Homomorphic Encryption, and Distributed Ledger Technology to ensure the confidentiality and integrity of data, security of private keys, low communication overheads, transparency and better scalability. It also facilitates building a verifiable authentic dataset to augment accurate analytics. The security analysis determines how SPAD mitigates the threats of privacy leakage and malicious collusion and facilitates privacy-preserving analytics. The experimental results, comprising execution time and memory usage, complement the efficient performance of the framework.**

***Index Terms*—Data integrity, data privacy, secure analytics, multi-party computation, distributed ledger technology**

I. Introduction

Secure Multi-Party Computation (SMPC) enables multiple parties to jointly compute a function without revealing their private data [1, 2]. However, conventional SMPC models face scalability issues [3], high computational overheads [4], lack of transparency [5], high latency [5], and have strong trust assumptions [6]. Another critical issue often neglected or needs to be addressed by the existing solutions is ensuring the security of the decryption/private keys.

Blockchain technology has been explored as a possible solution for SMPC challenges [7, 8, 9]. However, the proposed model suffers from high complexity, computational and communication overheads and an inability to handle large datasets. Further, the integration of blockchain and SMPC remains conceptually rich but technically underdeveloped [6]. Solutions like Carbyne Stack offer scalable SMPC applications but face challenges related to Kubernetes-based cloud security [10].

SPAD addresses the above gaps by combining lightweight cryptographic techniques with decentralized trust via blockchain. It mitigates the overhead of traditional SMPC by using Paillier Homomorphic Encryption (PHE) [11] for secure computations, Shamir's Secret-Sharing Scheme [12] for key protection, and Distributed Ledger Technology (DLT) (Ethereum Blockchain) for transparency and data integrity. SPAD is a secure-by-design framework without strong (impractical) assumptions or complex cryptographic primitives for security.

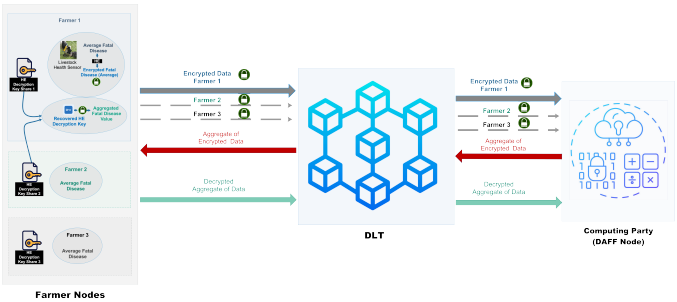

Fig. 1. SPAD Architecture

II. SPAD Framework Methodology

SPAD is designed for privacy-preserving multi-party analytics in a smart agriculture setting. As shown in Fig. 1, the framework consists of three key components: a) Farmer nodes (data owners): encrypt and share data via blockchain. b) Department of Agriculture, Fisheries and Forestry (DAFF)

node: computes encrypted aggregates without accessing raw data. c) DLT (Ethereum Blockchain): provides immutable data storage and facilitates transparent data sharing.

A. Data Flow and Key Security

Each session begins with a central Homomorphic (PHE) Key Generator producing a public-private key pair. The PHE private key is split into secret shares using Shamir's Secret-Sharing Scheme and distributed to farmer nodes. Each share is encrypted using the respective node's public key. Splitting the PHE private key into multiple shares ensures that no single party (stakeholder) has a complete view of the Key, and it remains secure. The security of the PHE private key improves with the increasing number of farmer nodes, as the threshold of secret shares required to recover the key will rise with the number of users. Farmer nodes encrypt their data with the PHE public key and publish the ciphertext on the blockchain. The DAFF node extracts encrypted values, computes an encrypted aggregate, and re-publishes it on the blockchain. Farmers then collaborate to reconstruct the decryption key using their secret shares and decrypt the aggregate for secure analytics. Each farmer node writes the decrypted aggregate value into a CSV file, along with the current timestamp and session identity (ID), to create a dataset. The farmer nodes also publish the decrypted aggregate value on the DLT so that the DAFF node can access it for further analytics.

B. Security Guarantees

SPAD protects farmers' data against unauthorized access by using PHE. The sharing of encrypted data using DLT smart contracts preserves the integrity of data and the overall process. Thus increasing users' trust in the framework. SPAD employs RBAC-by-design through restricted interfaces. Every node in the SPAD framework authenticates itself before logging into the respective interface. Moreover, the verifiable aggregate value extracted from the DLT and used for developing a dataset helps preserve the integrity of dataset values. Furthermore, as a fault tolerance measure, a mode of the decrypted aggregate values from the three farmers is computed and written into a new CSV file along with the timestamp and session ID.

The framework gets safer against collusion attacks with an increase in the number of users. In addition, the use of Elliptical Curve Cryptography (ECC) [13] with a key size greater than 256 bits is safe against quantum attacks as it would require a quantum computer with approximately 13 million physical qubits to brute force such a key [14]. Whereas, by the end of 2023, IBM's largest superconducting quantum computer had just over 1,000 physical qubits [15].

C. Performance Evaluation

As shown in Table I, the integration of DLT introduces a slight increase in the average (avg) execution (exec) time and memory (mem) usage, but overall, the impact remains negligible. Replacing RSA with ECC significantly improves execution time and memory consumption (comparison shown in Fig. 2). Comparing the efficiency of SPAD with other

TABLE I
PERFORMANCE ANALYSIS OF SPAD WITH DLT AND WITHOUT DLT

Operation	Avg Exec Time (ms)		Avg Mem Usage (Bytes)	
	No-DLT	DLT	No-DLT	DLT
Application launch	172.48	176.7	13926	14336
Keys and secret generation	280.01	283.49	4915	4976
Farmer nodes (initial exec)	33.95	83.64	32707	563472
DAFF node exec	0.1491	132.84	4505	194969
Farmer nodes (second exec)	64.24	163.16	628462	665053

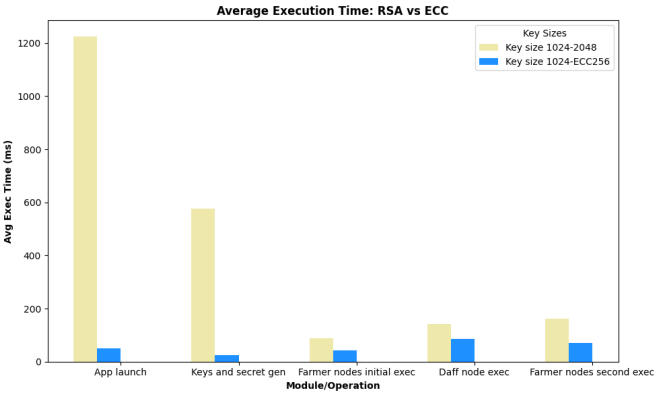

Fig. 2. Execution Time: RSA vs ECC

notable distributed data-sharing and SMPC frameworks, the average transaction commit time on Ethereum Blockchain was 4004 ms [16]. In another study, PrivySeC [17] exhibited transaction commit times of 484 ms and 314 ms for an encrypted and a plaintext data-sharing transaction, respectively. Similarly, a blockchain-based FL with an SMPC model verification framework [7] demonstrated a minimum processing time of 140 ms and 100 ms for deploying and verifying a transaction, respectively. In comparison, all the SPAD modules (for ECC ver) are faster.

III. CONCLUSION

SPAD provides a scalable, secure, and privacy-preserving analytics framework by integrating Paillier Homomorphic Encryption, Shamir's Secret-Sharing scheme, and DLT. It eliminates centralized trust assumptions while ensuring transparency, data integrity, and low communications overheads. Experimental evaluations demonstrate the efficiency of the ECC-based implementation. In the future, we aim to explore Fully Homomorphic Encryption schemes like Cheon-Kim-Kim-Song (CKKS) [18] and post-quantum cryptography solutions such as NTRUEncrypt [19] to enhance security.

ACKNOWLEDGMENT

The authors acknowledge the support of Food Agility CRC Ltd, funded under the Commonwealth Government CRC Program. The CRC Program supports industry-led collaborations between industry, researchers, and the community. The financial and in-kind support of Robert Bosch (Australia) Pty Ltd & Robert Bosch GmbH in completing this work is also acknowledged.

REFERENCES

[1] D. Evans, V. Kolesnikov, and M. Rosulek, "A pragmatic introduction to secure multi-party computation," *Foundations and Trends® in Privacy and Security*, vol. 2, no. 2-3, pp. 70–246, 2018. [Online]. Available: http://dx.doi.org/10.1561/3300000019

[2] N. Volgushev, M. Schwarzkopf, B. Getchell, M. Varia, A. Lapets, and A. Bestavros, "Conclave: secure multi-party computation on big data," in *Proc. 14^{th} EuroSys Conference 2019*, ser. EuroSys '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3302424.3303982

[3] Z. Guan, X. Zhou, P. Liu, L. Wu, and W. Yang, "A blockchain-based dual-side privacy-preserving multiparty computation scheme for edge-enabled smart grid," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 287–14 299, 2022.

[4] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, and Y. an Tan, "Secure multi-party computation: Theory, practice and applications," *Information Sciences*, vol. 476, pp. 357–372, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025518308338

[5] S. Wu, J. Li, F. Duan, Y. Lu, X. Zhang, and J. Gan, "The survey on the development of secure multi-party computing in the blockchain," in *Proc. 6^{th} International Conference on Data Science in Cyberspace (DSC)*, 2021, pp. 1–7.

[6] S. Sharma and W. K. Ng, "Scalable, on-demand secure multiparty computation for privacy-aware blockchains," in *Proc. Blockchain and Trustworthy Systems*, Z. Zheng, H.-N. Dai, M. Tang, and X. Chen, Eds. Singapore: Springer Singapore, 2020, pp. 196–211.

[7] A. P. Kalapaaking, I. Khalil, and X. Yi, "Blockchain-based federated learning with smpc model verification against poisoning attack for healthcare systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 12, no. 1, pp. 269–280, 2024.

[8] J. A. Khan, W. Wang, and K. Ozbay, "Believe: Privacy-aware secure multi-party computation for real-time connected and autonomous vehicles and micro-mobility data validation using blockchain—a study on new york city data," *Transportation Research Record*, vol. 2678, no. 3, pp. 410–421, 2024. [Online]. Available: https://doi.org/10.1177/03611981231180200

[9] G. Raja, Y. Manaswini, G. D. Vivekanandan, H. Sampath, K. Dev, and A. K. Bashir, "Ai-powered blockchain - a decentralized secure multiparty computation protocol for iov," in *Proc. IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 865–870.

[10] "Carbyne stack," 2022, Accessed on: Dec 20, 2024. [Online]. Available: https://carbynestack.io

[11] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Advances in Cryptology - EUROCRYPT '99*, ser. Lecture Notes in Computer Science, vol. 1592. Springer, 1999, pp. 223–238.

[12] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[13] D. Hankerson and A. Menezes, *Elliptic Curve Cryptography*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 1–2. [Online]. Available: https://doi.org/10.1007/978-3-642-27739-9_245-2

[14] B. Schneier, "Breaking 256-bit elliptic curve encryption with a quantum computer," 2022, Accessed on: Dec 22, 2024. [Online]. Available: https://www.schneier.com/blog/archives/2022/02/breaking-245-bit-elliptic-curve-encryption-with-a-quantum-computer.html

[15] D. Peev, "Quantum computing technology - a brief overview," in *Proc. 23^{rd} International Symposium on Electrical Apparatus and Technologies (SIELA)*, 2024, pp. 1–8.

[16] I. Makhdoom, M. Abolhasan, J. Lipman, D. Franklin, and M. Piccardi, "I2map: Iot device attestation using integrity map," in *Proc. 22^{nd} IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2023, pp. 1900–1907.

[17] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "Privysharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities," *Computers & Security*, vol. 88, p. 101653, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740481930197X

[18] J. H. Cheon, W. Cho, J. Kim, and D. Stehlé, "Homomorphic multiple precision multiplication for ckks and reduced modulus consumption," in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 696–710. [Online]. Available: https://doi.org/10.1145/3576915.3623086

[19] J. Hoffstein, N. Howgrave-Graham, J. Pipher, and W. Whyte, *Practical Lattice-Based Cryptography: NTRUEncrypt and NTRUSign*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 349–390. [Online]. Available: https://doi.org/10.1007/978-3-642-02295-1_11