Architectural Framework for an Enhanced Multi-Party Fully Homomorphic Encryption Scheme

Joshua Mamza[1], Idris Ismaila[1], Joseph Ojeniyi[1], Shafi'i Abdulhamid[2], Moses Noel[1] and Olusanjo Fasola[1]

[1]Department of Cyber Security, Federal University of Technology Minna, Niger State, Nigeria
[2]Department of Information Technology, Community College of Qatar, Qatar

Joshua.pg6759@st.futminna.edu.ng
Ismi.idris@futminna.edu.ng
ojeniyija@futminna.edu.ng
shafii.abdulhamid@ccq.edu.qa
moses.noel@futminna.edu.ng
sanjo.fasola@futminna.edu.ng

Abstract: The Common Vulnerability Scoring System (CVSS) depends on reliable vulnerability data from expert, but the current process of vulnerability score generation and transmission remain exposed to data manipulation and interception. Existing research work used supervised machine learning to automate CVSS scoring with up to 90% accuracy, but their plaintext-based approach lacked cryptographic protections, leaving it vulnerable to Man-in-the-Middle (MitM) attacks. Another research work introduced a homomorphic encryption-based framework that preserves data confidentiality during computation and offers moderate performance gains. However, their dependance on a single trusted aggregator, static key management, and absence of dynamic integrity threshold mechanisms left the system exposed if the aggregator's key or channel were compromised. An architectural framework for an Enhanced Multi-Party Fully Homomorphic Encryption Scheme (EMHES) was designed to combat Man-in-the-Middle (MitM) attacks targeting Vulnerability Score manipulation. By employing Homomorphic Encryption, the framework enables computations on encrypted vulnerability scores, ensuring confidentiality throughout their lifecycle. Key enhancements include integrating digital signatures to authenticate classified scores before encrypted transmission to cloud environments and verify the integrity of decrypted results post-processing. Digital signatures and regulatory oversight significantly strengthen security properties like non-repudiation, integrity, and confidentiality for cloud-based data computations. The EMHES architecture features a secure transmission channel with multiple security layers within the cloud service provider infrastructure. Additional security mechanisms include secure key management protocols, zero-knowledge proofs for integrity verification, and a resilient secure aggregation protocol designed to counter MitM attacks. From a computational analysis, baseline algorithms exhibit constant time complexity O(1), while the EMHES architecture operates with linear time complexity O(n). The result shows that EMHES provides superior security, integrity and performance on large datasets.

Keywords: Secure multiparty, Homomorphic encryption, Man-in-the-Middle attacks, Vulnerability score manipulation, EMHES

1. Introduction

The increasing interconnectivity of systems in serval domains that include healthcare, finance and cyber security, has highlighted the need of a secured data processing and sharing in terms of collaboration. Vulnerability Score (VS) plays a vital role in accessing the security position of these systems, organizations are guided in risk management approaches. Nonetheless, the manipulation of these scores poses significant threats through Man-in-the-Middle (MitM) attacks (Adablanu et al. 2024). VS systems such as the Common Vulnerability Scoring System (CVSS) are tools used for assessing and prioritizing security vulnerabilities using standard metrics used to evaluate the severity of vulnerability and potential impact. MitM attack significantly pose a threat by allowing malicious actors to intercept and alter vulnerability scores (Alanazi, Mahmood & Chowdhury 2023). Traditional encryption while deploying client confidentiality ensures decryption before computations are performed which creates vulnerability. Homomorphic encryption is a specialized encryption enables direct computations on an encrypted VS. Mathematical operations such as addition, multiplication is performed on encrypted file(Adablanu et al. 2024; Chen et al. 2024; Kamble, Jiet & Puri 2024).This paper proposes an architectural design for an Enhanced Multi-Party Fully Homomorphic Encryption Scheme (EMHES) to mitigate MitM attack against VS manipulation.

2. Background

2.1 Secure Multi-Party Computation (SMPC)

Secure Multi-party Computations (SMPC) is one of the important pillars in modern cryptography. Conversely, an Multi-party Computations (MPC) protocol allows a set of mutual distrust parties' i.e

$$P = (P_1, P_2, \ldots P_n) \quad (1)$$

To securely perform any computation or operation over their private input without revealing anything additional about their inputs. In most MPC protocols, the distrust is modelled by a centralized adversary A, who can corrupt and control parties during protocol execution. In perfect security, where A is a computationally unbounded adversary who can enforce the corrupt parties to behave arbitrary during protocol execution and where all the security guarantee are achieve without any error. Usually, the corruption capacity of A is modelled through a public known threshold *t.* where this is assumed. A can corrupt any parties up-to *t.*(Appan, Chandramouli & Choudhury 2023; Appan, Chandramouli & Choudhury 2025; Patil & Patra 2025). Private-preserving language model based on a SMPC addresses the privacy concerns associated with using large scale language model in Machine Learning as a Service (MLaaS). A scheme built on SMPC technology was propose to mitigate privacy leaking risks when processing sensitive data of a user. The proposed solution involves three non-colluding parties namely the data provider the model provider and the computing power provider. Compared to direct inference on large pre-trained models, this framework significantly improved inference speed by 1.55 to 6.25 times. The proposed solution goals stressed correctness privacy verifiability and efficiency (Song, Huang & Hu 2024; Yan et al. 2025; Zeng et al. 2025). SMPC are concerns with protocol execution coming under attack by an adversary which may corrupt one or more of the users to learn private information or cause the result said computation not to be correct. MPC are designed to prevent such kind of attack from being successful and can be proven mathematically to guarantee confidentiality, integrity and correctness (Pentyala et al. 2022; Yuan et al. 2021) . A secure and private multi-party deep learning through Differential Private – SMPC protocol which addresses security concerns about data privacy when it is train thereby achieving important improvement in communication and efficiency and speed compared to existing protocol (Das et al. 2025). SMPC perform an important role in block chain by allowing different parties to collaborate and compute on encrypted data without revealing their respective sensitive information. This process enhances privacy protection and data security block chain networks, it enables the block chain applications to execute complex task and smart contracts while safeguarding user privacy. SMPC delivers a secures and flexible data processing technique for block chain. It is used in the areas of finance, healthcare and supply chain (Bao et al. 2024; Ghanem & Moursy 2019; Zhou et al. 2021).

2.2 Vulnerability Scoring Manipulation

The management of vulnerabilities, which is a critical component of risk and resiliency efforts, typically is an ongoing process that involves the identification, classification, prioritization, and possibly remedy the vulnerabilities in devices likely to be targeted by adversaries to compromise network architecture and its components. Effective and efficient vulnerability management necessitates the allocation of resources such as time, personnel, and financial investment; thus, vulnerabilities should be prioritized according to their threat level (Jiang et al. 2025; Keskin et al. 2021; da Ponte, Rodrigues & Mattos 2023). Common Vulnerability Scoring System (CVSS) serves as a framework for assessing and communicating the severity of vulnerabilities and their potential impacts on devices. It provides a score that strongly echoes the severity of the vulnerability, enabling organizations and users to prioritize their remediall actions. The CVSS score is derived from metrics that evaluate various aspects of a vulnerability, precisely its impact on the confidentiality, integrity, and availability of the network system. (Tom & Kosacki 2023). A comparative analysis of large language models (LLMs) GPT and BERT for automated vulnerability scoring in cybersecurity field. It highlights that BERT outshines in understanding contextual relationships through bidirectional processing. GPT on the other hand, demonstrates strong generative capabilities but is more resource-intensive. The models' performance was evaluated on vulnerability classification tasks using the National Vulnerability Database (NVD) and finds that MediumBERT and SmallBERT outperform larger models like BERT and DistilBERT. In-addition, fine-tuning Generative Pre-trained Transformer (GPT) models significantly enhances their classification accuracy, particularly in specific vulnerability categories (Hashemi Chaleshtori & Ray 2023; Kalouptsoglou et al. 2024; Mirtaheri et al. 2025)

3. Related Work

(Sanon et al.2023 .) integrates federated learning (FL) with homomorphic encryption for predicting network traffic. This approach enhances privacy-preserving data analysis within a simulated environment of 5G. While the Cheon-Kim-Kim-Song (CKKS) encryption scheme requires increased computation time and inherent precision loss, the secure federated learning method demonstrates superior performance compared to traditional techniques. The results show that, despite the longer processing time required for the secure methods, ongoing advancements in homomorphic encryption are anticipated to improve efficiency in future applications.

A Multiparty Quantum Homomorphic Encryption (MQHE) scheme that allows multiple party perform quantum computations on their private data without requiring decryption, even in the faced with a dishonest server. The technique leverages measurement device independent quantum key Measurement-Divide-Independence Quantum Key Distribution (MDI-QKD) for secure generation and works with non-minimal states for error correction in computations. The proposed scheme outlines significant advantages in data protection during quantum computations, showing the application of quantum homomorphic encryption in secure multi party computation situations. Nonetheless, its complexity of implementation, potential computational overhead and assumptions about the behavior the server, limited scope of quantum operations present challenges that has to be addressed for deployment in several applications practically (Zhang et al., 2022; Li et al. 2025; Zhang et al. 2021). SMPCs perform an important role in block chain by allowing different parties to collaborate and compute on encrypted data without revealing their respective sensitive information. This process enhances privacy protection and data security block chain networks, it enables the block chain applications to execute complex task and smart contracts while safeguarding user privacy. SMPC delivers a secures and flexible data processing technique for block chain. It is used in the areas of finance, healthcare and supply chain (Bao et al. 2024) . Assessment testbed for cyber vulnerabilities by considering their result on business processes of an organization, beyond the standard CVSS metrics. It shows that Vulnerabilities with high CVSS scores do not always correlate with significant business impacts, urging decision-makers to prioritize vulnerabilities based on their true effects on organizational operations (Keskin et al. 2021). A comparative analysis of several FL techniques that emphasized their application in network traffic prediction. It discusses the significance of FL in data preservation privacy although collaborative model training among multiple organization. Accuracy and robustness are effectively identified, especially in the presence of attackers. The research focuses on application of homomorphic encryption alongside median aggregation. Additionally, the research also addresses the existing gaps regarding the comparative effectiveness of FL techniques creating way for improvement in the techniques of network traffic analysis (Sanon et al. 2024).

4. Research Methodology

The Enhanced Multi-Party Homomorphic Encryption Scheme (EMHES) has an implementation process that are structured and in multi-stage process comprising three main procedures: EMHES implementation stages, the time complexity analysis, and the space complexity analysis. This comprehensive approach ensures full functionality and performance of the designed EMHES.

4.1 EMHES Implementation Steps

1. Define, identification and retrieval of vulnerability score of

Retrieve Vulnerability Scores is equal to

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots x_{2n} \\ x_{31} & x_{32} & x_{33} & \dots x_{3n} \\ x_{m1} & x_{m2} & x_{m3} & \dots\ x_{mn} \end{bmatrix}, \qquad (2)$$

n = 12, m=104,976

2. An Architecture and flow chat have been developed to conceptualize the EMHES. Also, pseudocodes were been documents and analyzed to determine the efficiency of EMHES.

4.1.1 Time complexity analysis

The following steps outline the process for computing and analyzing the time complexity of the A = $\{A_i\ A_j A_k\}$ where the three key components are $A_i, A_j\ and\ A_k$ called Algorithm line number, Algorithm statement and Frequency count respectively. Where A equals sets of key components identified. A_i = 1, 2…n. Where $A_j =$ Statement 1, Statement 2…, Statement which represents the computations performed at each line of algorithm.

A_k = Count1, Count2 …, Count*n* which is computations performed at each Algorithm line. Where Count*i* is the number of times each operation is executed A = $\{A_i, A_j, A_k\}$. Also, let *f* frequency count of a computation. *f = 0 defines no computations performed, f = 1, defines computation is executed ones f= 1+1, defined computations is performed twice*, *f = n, n+1, defines a critical count in the loop during key generation and distribution*. The cumulative of the count F is represented as $\sum_{i=1}^{A_k}\{fA_k\}$. $A_k is\ total\ number\ of\ computations$, $fA_k\ is\ the\ frequency\ count\ in\ k-th\ computations$. Finally, *T* (n) = 6n +*x.*

where *T* is the algorithm time complexity which increases with the number of participants *n and x* is the constant.

4.1.2 Space complexity analysis

Sc is Space Complexity for Code per Character, fSccum is Cumulative frequency for code/Character, S_V is Space Complexity for Variable used in the algorithmic statement fSvcum is Cumulative count for space complexity for variable, S_{DC} is Space Complexity for Data Structure in the algorithmic statement

fS_{DC}cum is Cumulative count for space for data structure, S_{RA} is Space Complexity Recursive Algorithm for recursive calls and S_{RACUM} is Cumulative count for space complexity for recursive algorithm which is not applicable.

Also, A = $\{A_i\ A_j A_k\}$ where the three key components are $A_i, A_j\ and\ A_k$ called Algorithm line number, Algorithm statement and Frequency count respectively. Where A equals sets of key components identified. A_i = 1, 2…n, where A_j = Statement 1, Statement 2…, Statement which represents the computations performed at each line of algorithm. A_k = Count1, Count2 …, Countn which is computations performed at each Algorithm line

Note: Counti: is the number of times each operation is executed. A = $\{A_i, A_j, A_k\}$ Acum= $\{A_{SCCUM}, A_{SVCUM}, A_{SDCCUM}\}$

Therefore, the cumulative counts can be expressed as S_{CCUM} = $\{A_{SCCUM} = \sum_{i=1}^{K} Asci\}$, S_{VCUM} = $\{A_{VCUM} = \sum_{i=1}^{K} Asvi\}$, S_{DCCUM} = $\{A_{DCCUM} = \sum_{i=1}^{K} A_{DCCUMi}\}$. Where k, represents the number of operations associated with each algorithmic line.

This structured approach to the implementation of the EMHES ensures that both time and space complexities are thoroughly analyzed. Hence, an efficient and scalable system design and develop.

5. Result and Performance Analysis

5.1 The Complexity Analysis Came From the Depicted Flowchart

Figure 1 represents the Flow chat of the developed architecture where the process begins with the retrieval of the Vulnerability Score (VS), which is digitally signed to ensure integrity and authenticity. The system then determines the number of participating parties for the multi-party computation, after which cryptographic keys are generated and distributed among them. These keys are employed for encryption and decryption operations within the homomorphic encryption (HE) framework. A verification process is applied in which the system confirms whether the distributed key has been successfully generated and accepted by all parties. If agreed and key exchange are successful, the process continues otherwise, it iterates until all keys are correctly generated and distributed. Each party (P1 to Pn) privately retains its decryption key. The VS is then encrypted using HE to form Encrypted Vulnerability Score (EVS). Homomorphic operations are applied directly to the EVS, and the resulting ciphertext is transmitted to the Cloud Service Provider (CSP) where computations and storage are outsourced. During decryption stage, the digital signature of the ciphertext is first verified to confirm that the EVS has not been tampered during the transmission or processing operations. The system again determines the number of parties involved in the decryption phase and reconstructs the secret decryption key. If the required decryption keys are reconstructed and confirmed, the process proceeds; otherwise, it repeats until the decryption key can be successfully reconstructed. Lastly, the EVS is decrypted using the secret key, and the original VS is recovered with its digital signature intact, thereby maintaining both integrity and authenticity.

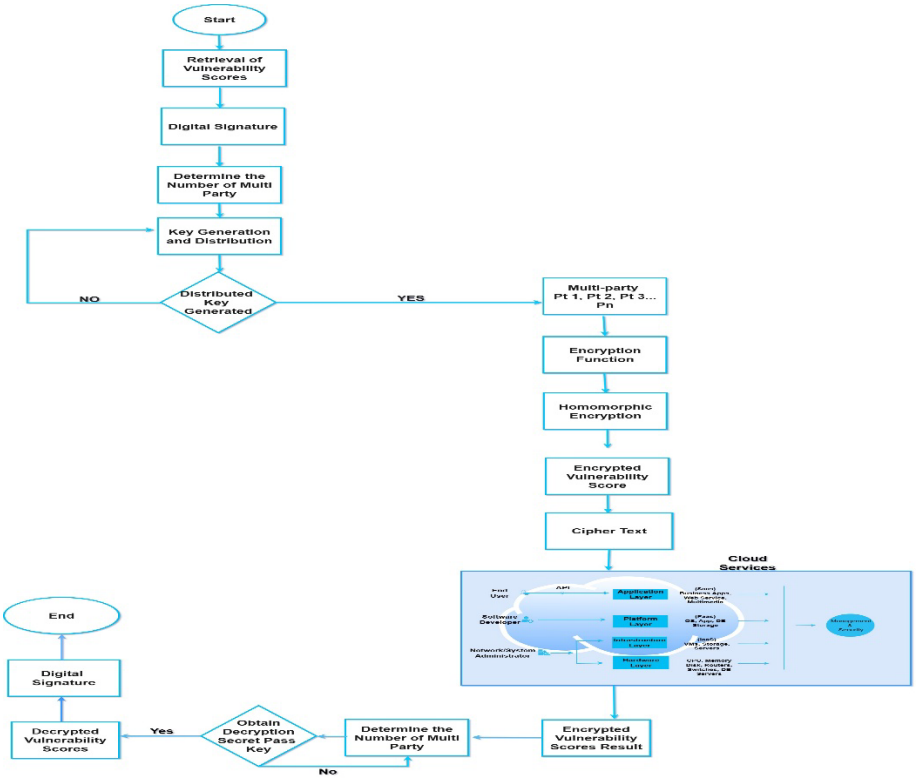

Figure 1: Flowchart for the Developed Architecture

5.2 Architecture for an Enhance Multiparty Homomorphic Encryption Scheme (EMHES)

Figure 2 represents EMHES. The architecture enables privacy-preserving computation and secure outsourcing of vulnerability scores (VS) to an untrusted cloud environment using homomorphic encryption. Vulnerability Scores (VS) are obtained from classified data, which are digitally signed to ensure integrity and authenticity. The digitally signed VS scores are handled by a trusted third-party government regulatory agency, where the data and keys are managed in a secure environment. Homomorphic encryption is initiated by the system where encryption keys are generated, and a multi-party computation (MPC) setup is established among participating entities (P1 to Pn). Each party receives and keep its own decryption key share, ensuring that no single party can decrypt the data unilaterally. The VS are encrypted using a homomorphic encryption scheme to produce Encrypted Vulnerability Scores (EVS), while preserving the ability to perform computations directly on the ciphertext. EVS is transmitted via a secured transmission channel to the Cloud Service Provider (CSP), where storage and computation are outsourced. Within the cloud environment, various layers such as application layer platform layer, Infrastructure layer etc. operates over the EVS without accessing to the plaintext VS data. These operation leverages on homomorphic function, allowing computations to be performed on ciphertext while data confidentiality is maintained. Note that the digital signatures attached to the EVS continue to protect against tampering, ensuring that any unauthorized alteration can be detected. During decryption stage, the EVS are returned via a secured transmission channel to the trusted environment where the digital signatures is verified and confirm that the EVS has not been altered during cloud processing or transmission. It then re-engages the multi-party decryption process, where the set of parties required for key reconstruction are determined and secret key are shares. The EVS is subsequently decrypted and decrypted VS retain their digital signatures, thereby preserving integrity and authenticity from end to end.

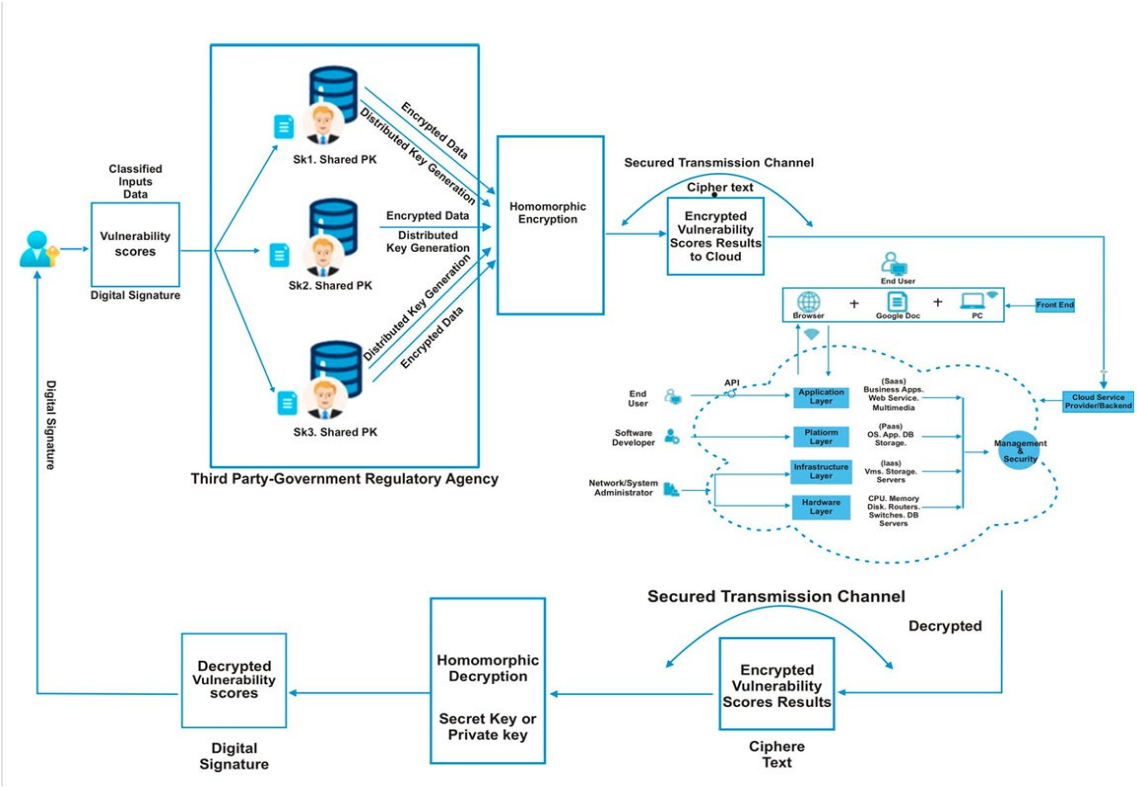

Figure 2: Architecture for an Enhance Multiparty Homomorphic Encryption Scheme (EMHES)

5.3 Result of the Analysis of Algorithm

Table 1 shows a comparative analysis of time complexities of algorithm for the designed architecture and baseline algorithms

Table 1: EMHES analysis

S/N	Author	Result of Time Complexity	Time complexity chat	Percentage Comparative Analysis
1	(Bao et al. 2024)-Base Line 1	O(1) + 32	Constant	0.32%
2	(Zhang et al. 2022)-Baseline 2	O(1) +101	Constant	0.98%
3	EMHES Architecture	6n+72	Linear complexity	6.52%

Table 1 shows the analysis of the algorithm od designed architecture (EMHES) for thetime complexity against existing algorithms. The EMHES algorithm exhibits a time complexity of 6n + 72 (O(n)), indicating that its computational effort grows linearly with the input size while the baseline algorithms from (Bao et al. 2024) and (Zhang et al. 2022) show constant time complexities of O(1) + 32 and O(1) + 101, respectively. Additionally, the EMHES architecture demonstrates a comparative analysis percentage of 6.52%, significantly higher than the 0.32% and 0.98% of the baseline algorithms. This indicates that the EMHES architecture offers enhanced security against Man-in-the-Middle (MitM) attacks, as it mandates increased computational effort from potential attackers, thereby improving overall security resilience.

Key assumptions:

- 1. General format of Constant Time Complexity of an algorithm (T_c):

$T_c = O(1) + K$

Where K is the units of time of algorithmic execution and O(1) is a Unit Big-O time of algorithmic execution

- 2. General format of Linear Time Complexity of an algorithm (T_L):

$T_L = O(n) + K$

Where O(n) is the n-time linear Big-O time of algorithmic execution and $K \leq n < \infty$

- 3. General format of Quadratic Time Complexity of an algorithm (T_Q):

$T_Q = O(n^2) + O(n) + K$

Where $O(n^2)$ is the n^2-time quadratic Big-O time of algorithmic execution

- 4. In this designed algorithmic analysis, let us assume n = K,

Where K is the greatest unit of time of algorithmic execution,

i.e., n = 101

Percentage (%) Comparative Analysis of the developed algorithm against the baseline work are given below.

For the Designed/Developed Algorithm (DA)

Note: Based on the assumption, n = 101

% PA Comparative Analysis

$(\%PA_{CA}) = \frac{Linear\ Time\ Complexity}{Quadratic\ Time\ Complexity}\ X\ 100$

$\%PA_{CA} = \frac{6n+72}{n^2+n+K}\ X\ 100$

$\%PA_{CA} = \frac{6(101)+72}{101^2+101+101}\ X\ 100$

$\%PA_{CA}$ = 6.52%

For the Baseline1 Algorithm (B1A),

% B1A Comparative Analysis

$(\%B1A_{CA}) = \frac{Constant}{Quadratic\ Time\ Complexity}\ X\ 100$

$\%\ B1A_{CA} = \frac{O(1)+32}{101^2+101+101}\ X\ 100$

$\%\ B1A_{CA} = \frac{1+32}{101^2+101+101}\ X\ 100$

$\%\ B1A_{CA} = \frac{33}{101^2+101+101}\ X\ 100$

$\%\ B1A_{CA}$ = 0.32%

For the Baseline2 Algorithm (B2A),

% B2A Comparative Analysis

$(\%B2A_{CA}) = \frac{Constant}{Quadratic\ Time\ Complexity}\ X\ 100$

$\%\ B2A_{CA} = \frac{O(1)+101}{101^2+101+101}\ X\ 100$

$\%\ B2A_{CA} = \frac{1+101}{101^2+101+101}\ X\ 100$

$\%\ B2A_{CA} = \frac{102}{101^2+101+101}\ X\ 100$

$\%\ B2A_{CA}$ = 0.98%

Hence $\%PA_{CA}$ = 6.52%, $\%\ B1A_{CA}$ = 0.32%, and $\%\ B2A_{CA}$ = 0.98%

Figure 3 shows the Time Complexity for The Designed Architecture. The bar chart which displays the relationship between Algo line number and number of frequency(ies) for 70 data points representing lines of code. It processes more input and shows that as the algorithm's line number increases. The proposed model (EMHES) incorporating Big O notation, demonstrates superior security, integrity, and performance compared to baseline models without Big O, and is better equipped to handle larger datasets.

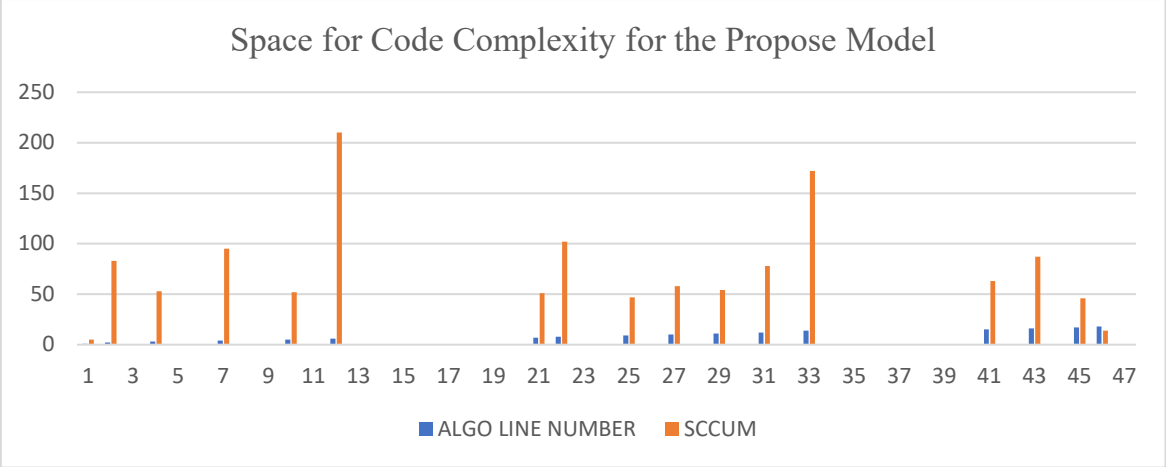

Figure 3: Time Complexity

C. Comparative analysis of space complexities of the algorithm

The comparative analysis of space complexities for the Enhanced Multi-Party Homomorphic Encryption Scheme (EMHES) against baseline algorithms is shown through a series of figures (Figures 4 to 5). These figures illustrate the space complexity of the designed architecture, showing metrics such as space for code complexity and data structure. Space Complexity Recursive Algorithm is not applicable in our algorithm as never recall itself with its on definition.

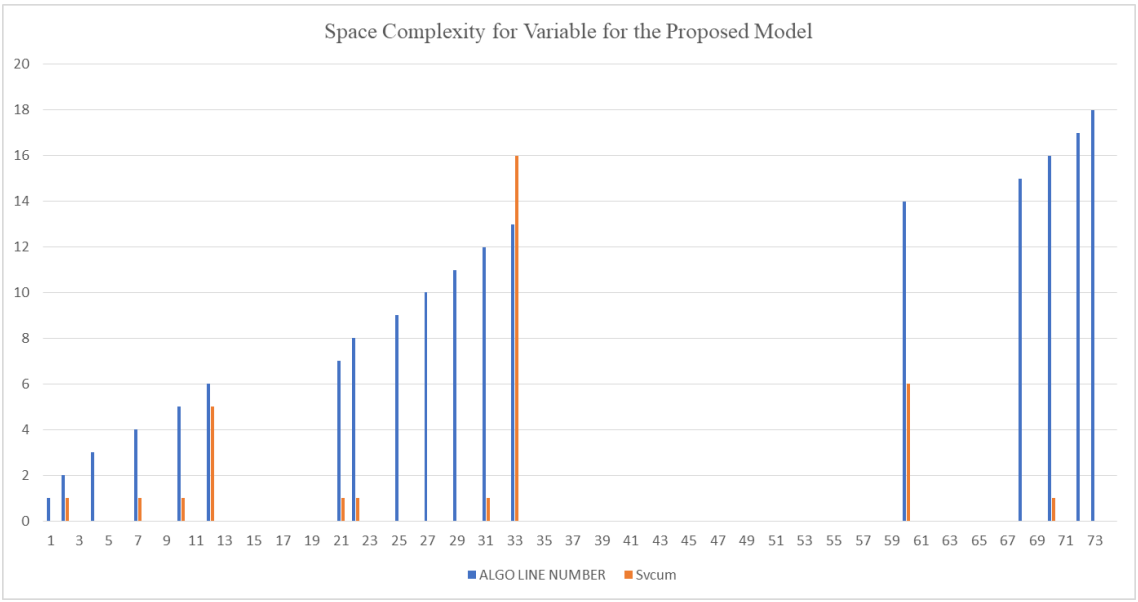

Figure 4. Space for Code Complexity

Figure 4 shows a Algo line number and Space for code complexity cumulative. Both plotted against an x-axis ranging from 1 to 73. The y-axis, representing space complexity cumulatively extends from 0 to 1400. S_{CCUM}, which represents the total memory space consumed by the algorithm during execution. This shows a significant spike around x-axis values 11, 33, and 60, at peak of 1300-1350 where x=33. These spikes indicate considerably higher memory resource usage, suggesting that certain algorithm computations, particularly due to cloud activities at x=33 and x=60, demand substantial computational resources and memory.

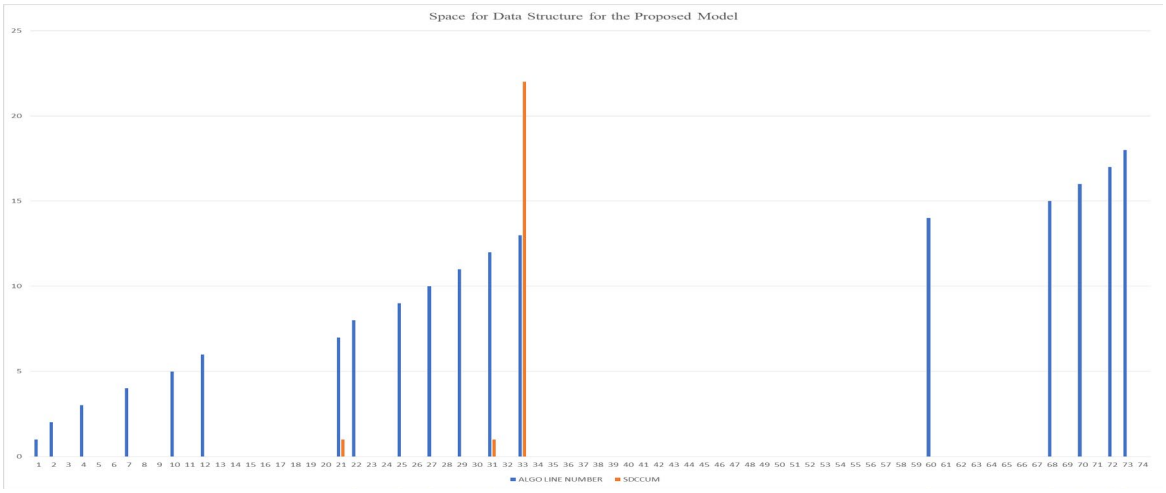

Figure 5: Space complexity for Data Structure EMHES

Fig. 5 illustrate the space complexity for data structures of the designed algorithm, with the x-axis representing Algorithm line number ranging from 1 to 73 and the y-axis showing space complexity ranging from 0 to 25. The Algo Line Number which indicate space complexity for data structures at various algorithm lines, while the cumulative space complexity data structure (S_{DCCUM}) values at each algorithm line. The Algo Line Number peaks around lines 59-73 where it has values of 14 and 18, whereas S_{DCCUM} has a significant peak algorithm line 33, reaching 22.

5.4 Discussion

This research provides comparisons of EMHES with existing frameworks of (Bao et al. 2024) "Direct Homomorphic Secured Multi-party computation (DHSMPC)", (Zhang et al. 2022) "Multi-party Quantum Homomorphic Encryption (MQHE)" and (Sanon et al. 2023) "Secure Federated Learning with HE". While DHSMPC focuses reduction of ciphertext size in for efficient storage and communication, it fails to address the complexity growth with larger input sizes. In contrast, EMHES demonstrates linear complexity, ensuring practical operation and enhancing resistance to MitM attacks as datasets increase. EMHES also offers immediate applicability in current cloud and regulatory infrastructures, where quantum solutions are not yet widely implemented. The work of (Sanon et al. 2023) highlights the trade-off between security and computational overhead in FL with homomorphic encryption, revealing that cryptographic protections can significantly extend processing times. EMHES mitigates this issue as it employs controlled linear time complexity, providing both scalability and security without excess overhead. Moreover, EMHES shows secure and tamper-resistant vulnerability evaluation, critical in regulatory settings where data integrity and authenticity must be maintained. EMHES provides a scalable solution that maintains performance and enhances attack resistance, making it ideal for large-scale, integrity-critical cloud-based vulnerability management, unlike existing studies that mainly emphasize compactness, quantum resilience, or privacy preservation.

6. Conclusion

In this study, EMHES was introduced as a robust architectural framework designed to mitigate MitM attacks on vulnerability score (VS) manipulation. Homomorphic encryption is integrated with advanced security measures namely digital signature and key management protocols, EMHES clearly preserve the confidentiality, integrity and authenticity of the VS through the process. In computational analysis, EMHES architecture maintains a linear time complexity contrasting significantly with the existing algorithm that shows constant time complexity. This attributes not only enhances security resilience against any potential attackers but also accommodate larger datasets, making it scalable. Additionally, the performance metrics based as assumed *n=100* demonstrated Comparative Analysis score of 6.25% for EMHES, evidently outperforming baseline algorithms. Furthermore, the detail analysis of the space complexity shows that the algorithm line of code requires significant computational resources especially during the cloud processing, showing areas of potential optimization. In conclusion, the EMHES framework offers a comprehensive, secure and efficient mechanism for collaborative processing of data is sectors where integrity and confidentially are important such as healthcare, finance and cybersecurity sectors.

Acknowledgements

All the technical inputs from team members and partners are all acknowledged and appreciated.

Ethics declaration: There is no ethical bridge or conflict in this research work necessitating ethical clearance and declaration. All actions complied with research guidelines as stipulated by the conference guidelines.

AI declaration: We employed POE to assist in conducting few literatures review. POE helped in identification of the relevant studies, but the selection and all the interpretation of the literature were performed by the authors.

References

Adablanu, S, Potter, K, Stilinki, D & Stilinski, D 2024, 'Homomorphic Encryption for Secure Cloud Computing Homomorphic Encryption for Secure Cloud Computing Homomorphic Encryption for Secure Cloud Computing'.

Alanazi, M, Mahmood, A & Chowdhury, MJM 2023, *SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues*, Elsevier Ltd, *Computers and Security*, vol. 125.

Appan, A, Chandramouli, A & Choudhury, A 2023, 'Network Agnostic Perfectly Secure MPC Against General Adversaries', in *Leibniz International Proceedings in Informatics, LIPIcs*, vol 281, Schloss Dagstuhl- Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing.

Bao, H, Yuan, M, Deng, H, Xu, J & Zhao, Y 2024, 'Secure multiparty computation protocol based on homomorphic encryption and its application in blockchain', Elsevier Ltd, *Heliyon*, vol. 10, no. 14.

Chen, Q, Li, H, Ariffin, S & Abdullah, NAS 2024, 'Overview of homomorphic encryption technology for data privacy', Conscientia Beam, *Review of Computer Engineering Research*, vol. 11, no. 3, pp. 130–139, viewed 6 September 2025, <https://ideas.repec.org/a/pkp/rocere/v11y2024i3p130-139id3955.html>.

Das, S, Ray Chowdhury, S, Chandran, N, Gupta, D, Lokam, S & Sharma, R 2025, 'Communication Efficient Secure and Private Multi-Party Deep Learning', Privacy Enhancing Technologies Symposium Advisory Board, *Proceedings on Privacy Enhancing Technologies*, vol. 2025, no. 1, pp. 169–183.

Ghanem, SM & Moursy, IA 2019, 'Secure Multiparty Computation via Homomorphic Encryption Library', Institute of Electrical and Electronics Engineers Inc., *Proceedings - 2019 IEEE 9th International Conference on Intelligent Computing and Information Systems, ICICIS 2019*, pp. 227–232.

Hashemi Chaleshtori, F & Ray, I 2023, 'Automation of Vulnerability Information Extraction Using Transformer-Based Language Models', Springer, Cham, *Lecture Notes in Computer Science*, vol. 13785 LNCS, pp. 645–665, viewed 7 September 2025, <https://link.springer.com/chapter/10.1007/978-3-031-25460-4_37>.

Jiang, Y, Oo, N, Meng, Q, Lim, HW & Sikdar, B 2025, 'A Survey on Vulnerability Prioritization: Taxonomy, Metrics, and Research Challenges', *Cornell University*, vol. 1, viewed 7 September 2025, <https://arxiv.org/pdf/2502.11070>.

Kalouptsoglou, I, Siavvas, M, Ampatzoglou, A, Kehagias, D & Chatzigeorgiou, A 2024, 'Vulnerability prediction using pre-Trained models: An empirical evaluation', IEEE Computer Society, *Proceedings - IEEE Computer Society's Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, MASCOTS*.

Kamble, A, Jiet, MM & Puri, C 2024, 'Homomorphic Encryption and its Applications in Multi-Cloud Security', Institute of Electrical and Electronics Engineers Inc., *7th International Conference on Inventive Computation Technologies, ICICT 2024*, pp. 1493–1499.

Keskin, O, Gannon, N, Lopez, B & Tatar, U 2021, *XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE Scoring Cyber Vulnerabilities based on Their Impact on Organizational Goals**.

Li, S, Li, S, Cai, X-Q, Cai, X-Q, Cai, X-Q, Wang, T-Y, Wang, T-Y & Wang, T-Y 2025, 'Secure multiparty computation for maximum and minimum values based on quantum homomorphic encryption', Optica Publishing Group, *Optics Express, Vol. 33, Issue 7, pp. 16263-16274*, vol. 33, no. 7, pp. 16263–16274, viewed 7 September 2025, <https://opg.optica.org/viewmedia.cfm?uri=oe-33-7-16263&seq=0&html=true>.

Mirtaheri, SL, Pugliese, A, Movahed, N & Shahbazian, R 2025, 'A comparative analysis on using GPT and BERT for automated vulnerability scoring', Elsevier B.V., *Intelligent Systems with Applications*, vol. 26.

Patil, S & Patra, A 2025, 'Perfectly-secure Network-agnostic MPC with Optimal Resiliency', Association for Computing Machinery (ACM), pp. 121–130, viewed 7 September 2025, </doi/pdf/10.1145/3732772.3733500?download=true>.

Pentyala, S, Railsback, D, Maia, R, Dowsley, R, Melanson, D, Nascimento, A & De Cock, M 2022, 'Training Differentially Private Models with Secure Multiparty Computation'.

da Ponte, FRP, Rodrigues, EB & Mattos, CLC 2023, 'CVEjoin: An Information Security Vulnerability and Threat Intelligence Dataset', Springer, Cham, *Lecture Notes in Networks and Systems*, vol. 661 LNNS, pp. 380–392, viewed 7 September 2025, <https://link.springer.com/chapter/10.1007/978-3-031-29056-5_34>.

Sanon, SP, Reddy, R, Lipps, C & Schotten, HD n.d., *Cross-Silo Horizontal Federated Learning Methods in Network Traffic Analysis*.

Sanon, SP, Reddy, R, Lipps, C & Schotten, HD n.d., *Secure Federated Learning: An Evaluation of Homomorphic Encrypted Network Traffic Prediction*.

Song, C, Huang, R & Hu, S 2024, 'Private-preserving language model inference based on secure multi-party computation', Elsevier B.V., *Neurocomputing*, vol. 592.

Tom & Kosacki 2023, 'BT_Prioritising_Vulnerability_Scan_Results'.

Yan, G, Zhang, Y, Guo, Z, Zhao, L, Chen, X, Wang, C, Wang, W, Meng, D & Hou, R 2025, 'Comet: Accelerating Private Inference for Large Language Model by Predicting Activation Sparsity', Institute of Electrical and Electronics Engineers Inc., *Proceedings - IEEE Symposium on Security and Privacy*, pp. 2827–2845.

Yuan, S, Shen, M, Mironov, I & Nascimento, ACA 2021, 'Practical, Label Private Deep Learning Training based on Secure Multiparty Computation and Differential Privacy', *Cryptology ePrint Archive*, viewed 7 September 2025, <https://eprint.iacr.org/2021/835>.

Zeng, W, Dong, Y, Zhou, J, Ma, J, Tan, J, Wang, R & Li, M 2025, 'MPCache: MPC-Friendly KV Cache Eviction for Efficient Private Large Language Model Inference', viewed 7 September 2025, <http://arxiv.org/abs/2501.06807>.

Zhang, JW, Chen, XB, Xu, G, Li, HJ, Wang, YL, Miao, LH & Yang, YX 2022a, 'A Secure Multiparty Quantum Homomorphic Encryption Scheme', Tech Science Press, *Computers, Materials and Continua*, vol. 73, no. 2, pp. 2835–2848.

Zhang, JW, Chen, XB, Xu, G, Li, HJ, Wang, YL, Miao, LH & Yang, YX 2022b, 'A Secure Multiparty Quantum Homomorphic Encryption Scheme', Tech Science Press, *Computers, Materials and Continua*, vol. 73, no. 2, pp. 2835–2848.

Zhang, J-W, 张静文, Chen, X-B, 陈秀波, Xu, G, 徐刚, Yang, Y-X & 杨义先 2021, 'Universal quantum circuit evaluation on encrypted data using probabilistic quantum homomorphic encryption scheme*', IOP Publishing, *Chinese Physics B*, vol. 30, no. 7, p. 070309, viewed 7 September 2025, <https://iopscience.iop.org/article/10.1088/1674-1056/ac003b>.

Zhou, J, Feng, Y, Wang, Z & Guo, D 2021, 'Using secure multi-party computation to protect privacy on a permissioned blockchain', MDPI AG, Sensors, vol. 21, no. 4, pp. 1–17.