A Verifiable AI-Augmented Homomorphic Secret Sharing Framework for Secure and Adaptive Distributed Computation

Mohammad Amin Khorzani
Dept. of math & Computer Science
Damghan University
Damghan, Iran
mohammadaminkhorzani@gmail.com

Majid Farhadi Sangkadehi
Dept. of math &Computer Science
Damghan University
Damghan, Iran
farhadi@du.ac.ir

***Abstract*—This work introduces a new framework for secure data processing in distributed environments through the combined use of classical secret sharing, strengthened symmetric encryption, verifiable computation, and adaptive monitoring based on deep learning. In this design, a confidential value is divided into independent components, and each component is separately encrypted and validated to prevent disclosure and manipulation. The verifiable computation layer enables the correctness of results to be confirmed without revealing any underlying information. In addition, an intelligent analysis mechanism based on computational vision models monitors encrypted representations and operational traces to detect abnormal or adversarial behavior during processing. The integration of these mechanisms forms a multilayer system that simultaneously ensures confidentiality, correctness, and adaptive oversight, providing a reliable foundation for cloud platforms, large-scale networks, and data-intensive applications.**

Keywords—Homomorphic Secret Sharing,AES-CBC Encryption, Zero-Knowledge Proof, Vision Transformer, Secure Data Sharing

I. INTRODUCTION

In today's data-driven era, where decision-making in science and industry increasingly relies on the analysis of sensitive information, the need for secure and trustworthy computation in untrusted environments has become critical. The widespread use of cloud infrastructures, federated learning systems, and distributed AI has led to the outsourcing of sensitive data to platforms that are not directly controlled by their owners. In such contexts, the three fundamental pillars of security—confidentiality, integrity, and verifiability—must be ensured simultaneously to maintain trust in outsourced computation [1].

To address these challenges, several privacy-preserving computation techniques have been developed, including fully homomorphic encryption (FHE) [4], secure multiparty computation (MPC) [3], and homomorphic secret sharing (HSS) [2]. Among these, HSS offers a practical and scalable approach that allows direct computation over shared data without revealing the underlying information. However, classical HSS only provides security against semi-honest adversaries and lacks cryptographic mechanisms to verify the correctness of outputs in the presence of malicious servers [3]. In contrast, while FHE guarantees strong theoretical privacy, its computational and communication overhead makes it impractical for real-time or IoT environments [4].

Recent studies, such as Dao et al. (2023), introduced the concept of verifiable homomorphic secret sharing (ve-HSS), where each server not only performs homomorphic computation but also generates a cryptographic proof (e.g., zk-SNARK) to verify correctness [5]. This approach effectively addresses the trust issue in outsourced computation and establishes the foundation for verifiable delegation. However, ve-HSS still lacks a dynamic and adaptive monitoring layer capable of detecting inference, poisoning, or reconstruction attacks in real time [6].

In parallel, a new direction of research has emerged in AI-driven secure data sharing [7]. This approach employs learnable encryption based on block-pixel operations and integrates it with the vision transformer (ViT) architecture to encrypt data while preserving discriminative features for classification. The work of Al Amin et al. (2025) demonstrates that such an approach maintains high robustness and accuracy—up to 94%—on encrypted medical datasets such as MRI and histopathology images, even under reconstruction and noise attacks [8].

Although ve-HSS and learnable encryption operate at different layers of the security stack—one ensuring provable cryptographic verifiability and the other enabling adaptive AI-based monitoring—their integration can form a comprehensive multi-layered framework for secure and intelligent computation. Motivated by this insight, this paper proposes AI-HSS (AI-Augmented Verifiable Homomorphic Secret Sharing), a unified architecture that combines the strengths of both paradigms to jointly achieve data confidentiality, computational integrity, and real-time anomaly detection.

The proposed architecture consists of four complementary layers:

1. Secure data delegation: splitting sensitive inputs into encrypted shares using threshold-based HSS to ensure data confidentiality.

2. Local homomorphic evaluation: executing target functions such as matrix multiplication or neural network inference on each share without inter-server communication.

3. Verifiable computation: generating compact cryptographic proofs (e.g., zk-SNARKs) to guarantee correctness of computation without revealing the underlying data.

4. AI-augmented monitoring: integrating learnable encryption with ViT to analyze encrypted data and detect abnormal behavior in real time.

In this design, learnable encryption serves as the adaptive bridge between encrypted visual inputs and secure inference, operating at the intersection of secure data delegation and AI-based monitoring. This combination enables privacy-preserving machine learning that is both cryptographically verifiable and dynamically intelligent.

Overall, AI-HSS bridges the gap between cryptographic theory and AI-based adaptability, establishing a new paradigm for trustworthy artificial intelligence in sensitive domains such as healthcare, IoT, and cloud computing.

II. Related Work

With the rapid expansion of federated learning, the Internet of Things, and cloud systems, the need for secure and trustworthy computation on sensitive data has led to the evolution of privacy-preserving computation techniques. This section reviews two complementary research directions: (1) homomorphic secret sharing and its verifiable extensions (ve-HSS), and (2) learnable encryption and AI-based secure data sharing.

2.1 Homomorphic Secret Sharing and Verifiable Evaluation

Classical secret sharing was introduced to divide a confidential secret into several independent shares [2]. Modern extensions, referred to as homomorphic secret sharing (HSS), generalize this concept to secure computation, enabling mathematical functions to be evaluated directly on encrypted shares, with the final result reconstructed from the combined outputs, while no individual server learns the original data [10].

Compared with MPC and FHE, HSS strikes a balance between computational simplicity, security, and scalability, making it suitable for distributed or resource-constrained environments [1][3][4]. However, classical HSS schemes are typically defined under the semi-honest model and provide no cryptographic guarantees for correctness in the presence of malicious evaluators [3].

Recent studies, particularly Dao et al. (2023), have sought to achieve sublinear multi-party computation using the Learning Parity with Noise (LPN) assumption, allowing efficient homomorphic evaluation in multi-party settings [5]. Later works, such as Choudhuri et al. (2023), introduced verifiable homomorphic secret sharing (ve-HSS), where each server, in addition to performing computation, produces a compact cryptographic proof verifying the correctness of its result [9]. This property enables publicly verifiable and trustworthy outsourced computation and forms the foundation of the Verifiable Computation layer in the proposed architecture.

2.2 Learnable Encryption and AI-Based Secure Data Sharing

In recent years, the field of learnable encryption has emerged as a bridge between cryptography and deep learning. In these methods, data are encrypted in a way that remains unintelligible to humans or adversaries while still allowing machine-learning models to perform training or inference directly on the encrypted domain. The work of Kiya and Ito (2022) introduced a Vision Transformer (ViT)-based framework using block-wise transformations such as pixel inversion, block scrambling, and channel shuffling [7]. This approach allows classification of encrypted data with minimal loss of accuracy.

Building on this idea, Al Amin et al. (2025) demonstrated that combining block-pixel operations with ViT can preserve up to 94 percent classification accuracy on encrypted medical datasets (MRI and histopathology) while maintaining robustness against reconstruction and noise attacks [8]. These approaches differ from HSS in that they rely not on algebraic cryptography but on model-compatible transformations that support adaptive learning. Although flexible, they generally lack formal protocol-level cryptographic guarantees.

This limitation provides strong motivation to integrate ve-HSS with learnable encryption. By combining the formal security of ve-HSS with adaptive AI-based monitoring, a unified framework can ensure both correctness of computation and real-time detection of abnormal behaviors or runtime attacks. In the proposed AI-HSS architecture, this synergy is realized through two complementary layers: the Verifiable Computation layer for correctness assurance, and the AI-Augmented Monitoring layer for intelligent encrypted analysis.

III. Proposed Method

3.1 Overview of the AI-HSS Architecture

The AI-HSS architecture integrates threshold secret sharing, verifiable computation, and adaptive monitoring to provide a secure foundation for distributed processing. It is designed to achieve confidentiality, integrity with verifiability, and dynamic anomaly detection through coordinated cryptographic and intelligent components.

Confidentiality is supported through Shamir secret sharing combined with independent AES-CBC encryption of each fragment. The sharing scheme ensures that no group of servers below the threshold can infer the underlying secret, while per-share encryption with unique keys and initialization vectors protects the system even if individual nodes are compromised.

Integrity and verifiability are maintained through a layered validation mechanism that includes SHA-256 commitments and zero-knowledge proofs. Each encrypted share is committed using a hash value to detect tampering and replay. During computation and reconstruction, nodes generate proofs of correct execution, enabling verification without exposing their private data.

Adaptive monitoring is achieved through an AI-based analysis module built on a Vision Transformer. This component examines encrypted representations and computation traces to detect abnormalities such as inconsistent responses, irregular timing, or deviations in proof behavior. When anomalies are identified, the system can isolate or down-weight affected shares, enhancing robustness against adversarial actions.

The overall architecture is organized into four coordinated layers: secure data sharing, local homomorphic evaluation, verifiable computation, and AI-based adaptive monitoring. Together, these layers enable AI-HSS to provide confidentiality, correctness, and adaptive resilience in distributed computation environments.

3.2 Threat Model and Security Assumptions

The system includes n independent servers, up to t−1 of which may be corrupted. Each server holds only its own share, and there is no direct inter-server communication.

3.3 Mathematical Definition of HSS

In the finite field F_q, a secret S is divided using a random polynomial of degree t−1:

$$P(x) = S + a_1x + a_2x^2 + \ldots + a_{t-1}x^{t-1} \bmod q \quad (1)$$

Each share is si = P(i), and reconstruction uses Lagrange interpolation:

$$s = \sum_{i=1}^{t} s_i \prod_{j \neq i} \frac{x_j}{x_j - x_i} mod\ q \quad (2)$$

Each share is further encrypted via AES-CBC to guarantee secure transmission.

3.4 Formal Security Analysis

Lemma 1: If reconstructing a degree-(t−1) polynomial in F_q without t shares is computationally infeasible, then AI-HSS is confidential.

Sketch of Proof: In game G_conf, adversary A has at most t−1 encrypted shares C_i = AES *ki*si), si = P(i). Reconstructing S=P(0) reduces to incomplete polynomial reconstruction, succeeding with probability $1/q^{t-1}$ Given AES-128 security, distinguishing AES from random is negligible.

Theorem 1: Assuming zk-SNARK based on Groth16 or PLONK under Discrete-Log or Ring-SIS hardness, no PPT adversary can produce y ≠ f(S) passing Verify() except with $\varepsilon \leq 2^{-\lambda}$

Proof Idea: By knowledge-soundness, any valid prover must know a witness w for f(S). With λ=128, verification succeeds in O(1) time with negligible error.

3.5 Security Assumptions and Game-Based Model

AI-HSS relies on three standard assumptions:

1. Sparse LPN hardness: recovering S from <t shares is equivalent to solving LPN with noise η ≥ 0.25 (success $\approx 2^{-\lambda}$

2. AES-128 resistance: adversary cannot distinguish AES from random.

3. zk-SNARK soundness with transparent setup (PLONK/Groth16): no forged proof accepted except negligible probability

Games:

G_conf – adversary guesses S with <t shares.

G_sound – adversary forges y' ≠ f(S) passing verification.

Advantage is bounded as:

Adv_AI-HSS(A) = |Pr[A wins G_conf] − Pr[A wins G_sound]| ⩽ $2^{-\lambda}$.

For λ = 128, global system security reaches 128-bit level.

3.6 Formal Algorithms

Algorithm 1 – Setup & Share:

Input: Secret S, threshold t, total servers n

Output: Encrypted shares $\{C_1,\ldots,C_n\}$

1. Choose F_q ($q=2^{128}$)

2. Generate

$$p(x) = s + \sum_{i=1}^{t-1} a_i x^i \quad (3)$$

3. For i in $\{1,\ldots,n\}$: $s_i=P(i)$; $C_i=AES_{Encrypt}(s_i, K_i)$; $h_i=SHA256(C_i)$

4. Send (C_i, h_i) to server i

Algorithm 2 – Local Evaluation:

Input: C_i, function f

Output: Ri, πi

1. $s_i=AES_{Decrypt}(C_i, K_i)$

2. $R_i=f(s_i)$ mod q

3. π_i=zkSNARK.Prove($R_i=f(s_i)$)

4. Return (R_i, π_i)

Algorithm 3 – Verify & Reconstruct:

Input: $\{R_i, \pi_i\}$ from t servers

Output: f(S)

1. For each i: zkSNARK.Verify(π_i)

2. If all valid: f(S)=LagrangeCombine($\{R_i\}$)

3. Return f(S)

3.7 Security Parameters

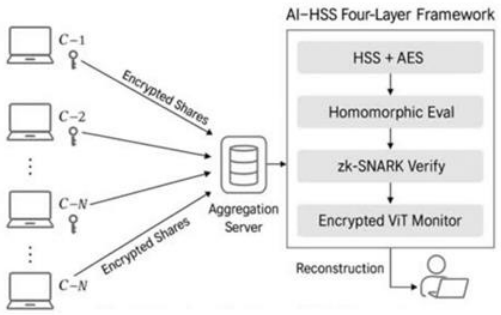
Parameter	Value	Description
Field size q	2^{128}	Matches AES-128 level
AES key length	128 bits	Per-share symmetric key
Servers n	5	Experimental topology
Threshold t	3	Two-fault tolerance
Hash function	SHA-256	Ensures data integrity

Table1. Summary of Core System Parameters

3.8 System Architecture Diagram

The following diagram illustrates the four-layer architecture of AI-HSS:

Fig1.AI-HSS four-layer architecture.



3.9 Complexity and Communication Analysis

The comparative positioning of our AI-HSS scheme against prior MPC and HSS frameworks is summarized in Table 2.

Scheme	Computation	Communication	Security	Adversary Model
Secure ML [1]	$O(n^2)$	High	Semi-honest	Honest
SPDZ [3]	$O(n^3)$	Very High	Malicious	Malicious
ve-HSS [9]	O(n log n)	Medium	Verifiable	Cryptographic
AI-HSS (ours)	O(n)	Low	AI-augmented + Malicious	Dual-level

Table2. Comparison of Secure Computation Schemes

3.10 Technical and Security Advantages

1. Multi-layer security: combines ve-HSS, AES, and zk-SNARK for both confidentiality and verifiability.

2. Reduced overhead: linear complexity and non-interactive layers improve scalability.

3. Adaptive anomaly detection: encrypted ViT identifies runtime deviations.

4. Cloud-ready design: modular and distributed for multi-party environments.

3.11.1 - Secret Sharing and Reconstruction Phase

The purpose of this phase is to transform a confidential secret S ∈ Fq into a set of verifiable, encrypted shares that can be safely distributed among n servers, and subsequently reconstructed only when at least t valid shares are available. The scheme integrates Shamir's Secret Sharing, AES-CBC encryption, SHA-256 hashing, and zk-SNARK-based verification, all within the AI-HSS framework to ensure confidentiality, integrity, and verifiability.

3.11.2 distribution phase of the proposed secret sharing scheme

A random polynomial of degree (t−1) over Fq is generated:

$p(x) = s + \sum_{i=1}^{t-1} a_i x^i$ (4) , with random coefficients $a_j \in F_q$.

Each participant i receives share s_i = P(i), then encrypts it using AES-CBC with key K_i and IVi.

Each encrypted share Ci is authenticated by SHA-256 to produce h_i, forming final triplets (C_i, h_i, metai).

Step	Operation
1	Choose random coefficients $a_1, a_2, \ldots a_{t-1}$ uniformly from F_q.
2	Define $p(x) = s + \sum_{i=1}^{t-1} a_i x^i$
3	For each participant i = 1,2,..,.n: compute $S_i = P(i)$.
4	Generate random IV_i for each share.
5	Encrypt: C_i = AES_CBC_Enc(K_i, (s_i || ID_i); IVi).
6	Compute hash: hi = SHA-256(C_i || IV_i || ID_i).
7	Construct metadata metai = (IV_i, IDi, ts).
8	Return {(Ci, hi, metai)}i=1..n.

Table3. Distribution secret sharing algorithm

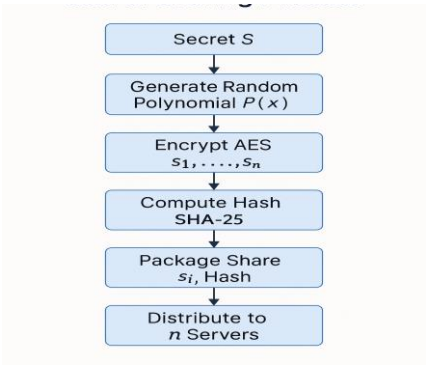


Fig2. Secret Sharing Phase

3.11.3 Secret Reconstruction Phase

The goal is to recover the original secret S (or f(S)) from at least t verified shares while filtering corrupted or malicious ones via zk-proofs and AI-based monitoring. The process includes verification, decryption, proof checking, and final Lagrange interpolation.

Step	Operation
1	Verify hash: hi ?= SHA-256(C_i || IV_i || ID_i).
2	Decrypt Ci using AES-CBC: si = AES_CBC_Dec(K_i, C_i; IV_i).
3	If zk-proof exists, check Verify(πi)=1.
4	Remove or down-weight anomalous shares based on monitoring flag αi.
5	If |T| < t, abort reconstruction.
6	Compute Lagrange coefficients λ_i for i ∈ T.
7	Reconstruct $s = \sum_{i \in t} s_i \lambda_i \, (mod q)$(5)
8	Return (S or f(S), audit log).

Table4.Reconstruction secret sharing algorithm

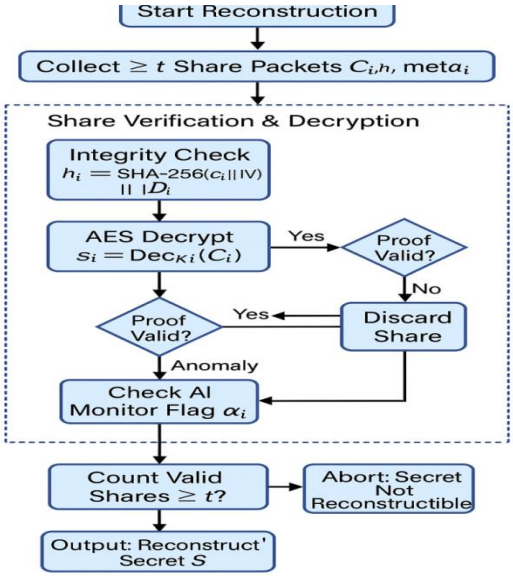


Fig3. Secret Reconstruction Phase

The reconstruction workflow, including integrity checks, decryption, proof verification and AI-based anomaly detection, is summarized in Fig. 3.

3.11.4 Correctness and Complexity Analysis

A concise summary of the main operational properties of the system is presented in Table 5.

Aspect	Description
Correctness	Any t valid shares reconstruct a unique S due to Lagrange interpolation.
Fault Tolerance	Invalid or malicious shares are filtered via hash, proof, and monitoring.
Computational Cost	Reconstruction requires $O(t^2)$ time; verification O(t).
Communication Cost	Only t valid packets required; cost is linear in t.

Table5. performance analysis

3.12 Experimental analysis

To evaluate the performance of the proposed scheme, experiments were conducted for different values of n with a fixed threshold of t=3. The main stages include sharing, local evaluation, verification, and the lightweight analytical layer. Execution times were measured over 20 independent runs and reported as average values with standard deviation.

The results show that the cryptographic core of AI-HSS is highly efficient, with total latency remaining in the range of 0.3 to 2.1 milliseconds even at larger scales. The analytical processing layer, responsible for monitoring and basic inference, adds an average overhead of about 5 milliseconds and integrates into the workflow without affecting the correctness of reconstruction.

As summarized in Table 6, the system achieves 100 percent successful reconstruction in all configurations, and the total execution time grows linearly and predictably with the number of servers. Overall, the findings indicate that the proposed framework offers stable, low-latency performance and can incorporate analytical components while maintaining reliability and efficiency.

n	t	share_ms (avg±std)	eval_ms (avg±std)	verify_ms (avg±std)	monitor_ms (avg±std)	total_ms (avg±std)
5	3	0.131±0.070	0.156±0.038	0.034±0.008	5.12±0.41	5.441±0.44
10	3	0.207±0.028	0.287±0.046	0.032±0.002	5.12±0.43	5.646±0.45
20	3	0.377±0.030	0.541±0.058	0.033±0.002	5.11±0.42	6.061±0.47
50	3	0.871±0.016	1.266±0.042	0.034±0.002	5.13±0.39	7.301±0.50

Table6. Performance Evaluation of the Proposed Scheme

conclusion and Future Work

This paper introduced AI-HSS, a structured framework that combines threshold secret sharing, symmetric encryption, verifiable computation, and adaptive monitoring to enable secure and trustworthy distributed processing. The architecture ensures confidentiality through Shamir-based sharing and per-share AES-CBC protection, while integrity and correctness are maintained via hash commitments and zero-knowledge verification. The inclusion of an AI-driven monitoring component further strengthens resilience by identifying irregular or potentially malicious behavior during computation.

Future directions include optimizing the verification layer, evaluating alternative lightweight monitoring models, and extending the architecture to larger and more heterogeneous network settings. Additional investigation into different cryptographic assumptions and threshold parameters may further improve efficiency and robustness. Overall, AI-HSS provides a consolidated foundation for secure, verifiable, and adaptively monitored distributed computation.

REFERENCES

[1] P. Mohassel and Y. Zhang, "SecureML: A System for Scalable Privacy-Preserving Machine Learning," 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 19-38, doi: 10.1109/SP.2017.12.

[2] A. Shamir, “How to share a secret,” Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979, doi: 10.1145/359168.359176.

[3] S. Wagh, D. Gupta, and N. Chandran, “SecureNN: 3-Party Secure Computation for Neural Network Training,” Proc. Privacy Enhancing Technologies*, vol. 2019, no. 3, pp. 26–49, 2019, doi: 10.2478/popets-2019-0035.

[4] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in Proc. 41st ACM Symposium on Theory of Computing (STOC), 2009, pp. 169–178, doi: 10.1145/1536414.1536440.

[5] Q. Dao, Y. Ishai, A. Jain, and H. Lin, “Multi-Party Homomorphic Secret Sharing and Sublinear MPC from Sparse LPN,” in Advances in Cryptology – CRYPTO 2023, Springer, 2023, doi: 10.1007/978-3-031-38545-2_11.

[6] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” in *Advances in Cryptology — EUROCRYPT 2003, LNCS vol. 2656, Springer, 2003, doi: 10.1007/3-540-39200-9_26.

[7] H. Ito, Y. Kinoshita, and H. Kiya, “Image Transformation Network for Privacy-Preserving Deep Neural Networks and Its Security Evaluation,” in *Proc. IEEE 9th Global Conference on Consumer Electronics (GCCE), 2020, doi: 10.1109/GCCE50665.2020.9292064.

[8] Amin, A., Hasan, K., Ullah, S., & Hong, L. (2025, February). AI-Driven Secure Data Sharing: A Trustworthy and Privacy-Preserving Approach. In 2025 International Conference on Computing, Networking and Communications (ICNC) (pp. 174-179). IEEE.

[9] Choudhuri, A. R., Goel, A., Hegde, A., & Jain, A. (2024). Homomorphic secret sharing with verifiable evaluation. IACR Cryptology ePrint Archive.

[10] Boyle, E., Gilboa, N., & Ishai, Y. (2016). *Function secret sharing: Improvements and extensions*. In

Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (pp. 1292–1303). https://doi.org/10.1145/2976749.2978429.