International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2026/08 Volume 13, Issue 1, January – February (2026)

RESEARCH ARTICLE

HE-Cloud: A DSL-Driven Homomorphic Encryption Framework with ZKP and ORAM for Privacy-Preserving Data Analytics

Tuan Nguyen Kim
Phenikaa School of Computing, Phenikaa University, Duong Noi, Hanoi, Vietnam.
✉ tuan.nguyenkim@phenikaa-uni.edu.vn

Ha Nguyen Hoang
University of Sciences, Hue University, Hue, Vietnam.
nguyenhoangha@hueuni.edu.vn

Son Doan Trung
Phenikaa School of Computing, Phenikaa University, Duong Noi, Hanoi, Vietnam.
son.doantrung@phenikaa-uni.edu.vn

Lam Tran Nguyen
Department of Science and Technology of Da Nang City, Danang, Vietnam.
lamnho1998.98@gmail.com

Received: 25 September 2025 / Revised: 21 December 2025 / Accepted: 06 January 2026 / Published: 26 February 2026

Abstract – Cloud computing has become a vital platform for large-scale data analytics, yet it poses significant privacy challenges when handling sensitive information, especially in healthcare and financial domains. Homomorphic Encryption (HE) enables computation on encrypted data, providing strong privacy guarantees, but traditional HE frameworks lack efficient query representation, do not protect query patterns, and cannot prove correctness of cloud-side computations. This paper proposes HE-Cloud, an integrated privacy-preserving framework that combines DSL-driven query compilation, HE, Zero-Knowledge Proofs (ZKP), and Oblivious RAM (ORAM). Our framework allows clients to express high-level analytical queries, securely executes them on encrypted data, protects query access patterns via ORAM, and returns verifiable results through ZKP. A proof-of-concept implementation using the Pima Diabetes dataset demonstrates feasibility: Average glucose computations can be performed entirely on encrypted data with sub-second latency for homomorphic operations and minimal accuracy loss (approximately 0.001). Scalable secure analytics, extendable to larger datasets and machine learning tasks.

Index Terms – Homomorphic Encryption, Privacy-Preserving Cloud Computing, Secure Data Analytics, Encrypted Medical Data, Zero-Knowledge Proof.

1. INTRODUCTION

Cloud computing has become a dominant platform for storing and processing large-scale data due to its scalability, flexibility, and cost efficiency. In many application domains, especially finance, healthcare, and government services, organizations increasingly rely on cloud infrastructures to perform data analytics and decision support. However, these domains often involve highly sensitive data, such as personal information, financial records, or transactional histories, which raises serious concerns about data privacy and confidentiality when outsourcing computation to third-party cloud servers.

To address these concerns, a wide range of privacy-preserving techniques have been proposed. Homomorphic Encryption (HE) [1-2] enables computation directly on encrypted data, ensuring that data confidentiality is preserved even during processing. Zero-Knowledge Proofs (ZKP) [3-4] allow a server to prove the correctness of computation results without revealing any additional information about the underlying data. Oblivious RAM (ORAM) [5-6] hides access patterns during data retrieval, preventing adversaries from inferring sensitive information based on query behavior. Each of these techniques targets a specific privacy or security dimension and has been extensively studied in isolation.

Despite their strong theoretical guarantees, deploying these techniques in practical cloud-based data analytics systems remains challenging. HE typically incurs high computational

RESEARCH ARTICLE

overhead and requires careful parameter tuning to balance accuracy and performance. ZKP introduces additional costs for proof generation and verification, especially for complex computations. ORAM significantly increases latency and communication overhead due to data shuffling and indirect access mechanisms. More importantly, existing studies often apply these techniques independently or focus on a single protection aspect, such as data confidentiality or result correctness, without considering their combined impact on system performance and usability.

As a result, there is a lack of unified frameworks that systematically integrate HE, ZKP, and ORAM into a single cloud-based analytics architecture. Even fewer works provide a quantitative analysis of the trade-offs between security and performance when these techniques are combined under different configurations. In real-world deployments, especially in financial environments, system designers rarely require the strongest possible protection in all situations. Instead, they must choose appropriate security mechanisms based on data sensitivity, trust assumptions, and performance constraints. However, current literature provides limited guidance on how to make such choices in practice.

1.1. Problem Statement

Given a cloud-based data analytics scenario involving sensitive financial data, the following challenges remain insufficiently addressed:

- How to design a unified architecture that integrates HE, ZKP, and ORAM while remaining practical for real-world deployment.
- How to support expressive analytical queries on encrypted data without exposing data content, computation correctness, or query behavior.
- How to quantitatively evaluate the performance-security trade-offs introduced by different combinations of HE, ZKP, and ORAM.
- How to provide practical guidance for selecting an appropriate security configuration based on application requirements rather than assuming a one-size-fits-all solution.

This work aims to address these challenges by proposing and evaluating a configurable cloud-based framework for privacy-preserving data analytics in the banking domain.

1.2. Contributions of the Study

The main contributions of this paper are summarized as follows:

- Unified Architecture: We propose a cloud-based data processing architecture that integrates Homomorphic Encryption, Zero-Knowledge Proofs, and Oblivious RAM within a single framework, supported by a dedicated HE-DSL Compiler [7-8] for query optimization.
- Configurable Security Design: We define three security configurations: HE only, HE with ZKP, and HE with ZKP and ORAM, allowing flexible adaptation to different security and performance requirements.
- Practical Query Support: We demonstrate how common statistical queries on banking data can be expressed using a domain-specific language and efficiently executed on encrypted data.
- Quantitative Trade-off Analysis: Through experimental evaluation on a real-world banking dataset, we provide a detailed analysis of processing time, communication overhead, and accuracy across different security configurations.
- Deployment-Oriented Insights: Based on the experimental results, we offer practical recommendations for selecting suitable security configurations in cloud-based financial data analytics.

The remainder of this paper is organized as follows. Section 2 reviews related work on privacy-preserving cloud data analytics using HE, ZKP, and ORAM. Section 3 presents the proposed system architecture and processing workflow. Section 4 describes the experimental setup, including the dataset, configurations, and deployment environment. Section 5 reports and analyzes the experimental results. Section 6 discusses the security implications, limitations, and practical considerations of the proposed approach. Finally, Section 7 concludes the paper and outlines directions for future research.

2. RELATED WORK

Privacy-preserving data analytics in cloud environments has attracted significant research attention in recent years, especially for applications involving sensitive data such as financial records, healthcare information, and personal data. Existing studies can be broadly categorized into four main research directions: (i) Homomorphic encryption-based secure computation, (ii) Zero-knowledge proof-based result verification, (iii) Oblivious data access mechanisms using ORAM or PIR, and (iv) Integrated frameworks combining multiple privacy-enhancing technologies. This section reviews representative and recent works in each category, highlighting their methodologies, advantages, and limitations.

2.1. Homomorphic Encryption-Based Secure Data Analytics

Gentry [9] introduced the first fully homomorphic encryption (FHE) scheme, enabling arbitrary computation on encrypted data. While groundbreaking, the original construction was computationally expensive, limiting its practical deployment. Subsequent research focused on improving efficiency and

RESEARCH ARTICLE

practicality.

Cheon et al. [10] proposed the CKKS scheme, which supports approximate arithmetic on real numbers and is well suited for statistical analysis and machine learning workloads. CKKS significantly reduces computational overhead compared to early FHE schemes but introduces approximation errors and requires careful parameter selection to balance precision and performance.

Several studies applied HE to cloud data analytics. Dowlin et al. [11] demonstrated encrypted neural network inference using leveled HE, showing feasibility but at high computational cost. Kim et al. [12] applied CKKS to financial time-series analysis, achieving confidentiality but without mechanisms to verify result correctness or protect access patterns.

Alam et al. [13] explored HE-based secure banking data processing on the cloud, focusing on encrypting customer data and performing basic statistical queries. However, their approach does not address query privacy or provide guarantees against incorrect or malicious computation by the cloud server.

These works show that HE effectively protects data confidentiality during computation, but most HE-only solutions lack mechanisms for result verification and query-pattern protection, which are critical in untrusted cloud environments.

2.2. Zero-Knowledge Proofs for Result Correctness

Zero-knowledge proofs (ZKP) enable a prover to convince a verifier that a computation is correct without revealing any additional information. zk-SNARKs, introduced by Ben-Sasson et al. [14], provide succinct proofs with fast verification, making them attractive for outsourced computation.

Pinocchio [15] presented a practical zk-SNARK framework for verifying arithmetic circuits, but its applicability is limited by circuit size and setup complexity. Zhao et al. [6] combined ZKP with HE to verify computations on encrypted medical data, demonstrating improved trust in cloud results. However, their study does not evaluate performance overhead in large-scale or high-frequency query scenarios.

Wang et al. [17] proposed a ZKP-based verification mechanism for cloud analytics workflows, focusing on correctness guarantees. While effective for ensuring result integrity, their approach assumes plaintext computation and does not address data confidentiality.

Overall, ZKP-based approaches provide strong correctness guarantees but introduce additional computational and communication overhead. Moreover, many existing works consider ZKP independently and do not analyze its combined impact with HE and ORAM in practical deployments.

2.3. Oblivious Data Access Using ORAM and PIR

Protecting query behavior is another critical aspect of privacy-preserving cloud analytics. Goldreich and Ostrovsky [18] introduced Oblivious RAM (ORAM) to hide access patterns by reshuffling and re-encrypting data blocks. Path ORAM, proposed by Stefanov et al. [19], significantly reduces ORAM complexity and has become one of the most widely adopted constructions.

Wang et al. [20] applied ORAM to secure cloud database queries, demonstrating effective access-pattern hiding but with notable latency overhead. Chor et al. [21] explored Private Information Retrieval (PIR) as an alternative, enabling clients to retrieve records without revealing query indices, but PIR-based solutions often suffer from high communication costs.

While ORAM and PIR effectively protect query privacy, they are typically studied in isolation or combined with symmetric encryption. Their integration with HE and ZKP for encrypted analytics remains underexplored, particularly in terms of performance trade-offs.

2.4. Integrated Privacy-Preserving Frameworks

A smaller number of studies attempt to integrate multiple privacy-enhancing technologies into unified frameworks. Zhao et al. [22] proposed a secure cloud analytics framework combining HE and ZKP, focusing on correctness and confidentiality. However, their framework does not address query-pattern leakage.

Patil and HimaBindu [23] proposed a cloud encryption mechanism based on CuLOA and deep learning models to optimize key generation. While improving key management efficiency, their approach does not support secure computation on encrypted data.

More recent works explore hybrid approaches for privacy-preserving analytics but often lack systematic evaluation of the trade-offs between security and performance when combining HE, ZKP, and ORAM. In particular, few studies focus on financial data scenarios, where both strong privacy guarantees and practical efficiency are required.

In summary, existing literature provides valuable building blocks for privacy-preserving cloud analytics. However, there remains a gap in unified, configurable frameworks that integrate HE, ZKP, and ORAM and quantitatively analyze their combined impact on performance and security in real-world applications.

2.5. Summary of Related Work

The summary in Table 1 highlights that existing studies typically focus on only one or two aspects of privacy-

RESEARCH ARTICLE

preserving analytics, such as data confidentiality, result correctness, or query-pattern protection. While these approaches provide strong guarantees in isolation, they often incur significant overhead and lack a unified design that balances security and performance. Moreover, few works offer a configurable framework that allows system designers to select appropriate protection levels based on application requirements. These limitations motivate the proposed approach, which integrates HE, ZKP, and ORAM within a single architecture and enables systematic trade-off analysis.

Table 1 Summarizes the Key Characteristics, Advantages, and Limitations of the Reviewed Studies

Ref	Main Technique	Application Domain	Advantages	Limitations
[9]	FHE	General	Arbitrary computation on encrypted data	Extremely high computational cost
[10]	CKKS	General analytics	Efficient approximate arithmetic	Approximation error, parameter tuning
[11]	HE	ML inference	Encrypted model execution	High latency
[12]	HE	Financial data	Data confidentiality	No correctness or query privacy
[13]	HE	Banking	Secure basic analytics	No ZKP or ORAM
[14]	zk-SNARK	General	Succinct proof, fast verification	Trusted setup
[15]	ZKP	Secure computation	Strong correctness guarantees	Circuit complexity
[16]	HE + ZKP	Healthcare	Confidentiality + correctness	Performance not analyzed
[17]	ZKP	Cloud analytics	Result verification	No data confidentiality
[18]	ORAM	Storage	Access-pattern hiding	High overhead
[19]	Path ORAM	Cloud storage	Reduced ORAM complexity	Increased latency
[20]	ORAM	Cloud DB	Query privacy	Communication overhead
[21]	PIR	Data retrieval	Private queries	High bandwidth cost
[22]	HE + ZKP	Cloud analytics	Confidentiality + correctness	No query-pattern protection
[23]	Key optimization	Cloud security	Efficient key management	No encrypted computation
Ours	HE + ZKP + ORAM	Banking analytics	Full confidentiality, correctness, and query privacy	Higher computational and communication cost

3. ROPOSED SYSTEM

This section proposes HE-Cloud, a privacy-preserving framework that integrates DSL, HE, ZKP, and ORAM. HE-Cloud enables high-level analytical queries to be compiled into homomorphic circuits, securely executed on encrypted data, and verified for correctness without revealing query patterns or plaintext values. The subsequent subsections present its architecture, workflow, and proof-of-concept using the Pima Diabetes dataset.

3.1. Overall Architecture

The proposed HE-Cloud framework comprises four core components (Figure 1): (i) Domain-Specific Language (DSL): (ii) Homomorphic Encryption (HE); (iii) Oblivious RAM (ORAM)/Private Information Retrieval (PIR); and (iv) Zero-Knowledge Proofs (ZKP), that work together to enable secure and verifiable cloud analytics.

(i) DSL: Provides a high-level abstraction for expressing analytical queries. User queries are compiled into optimized homomorphic circuits, eliminating the need for developers to manually handle cryptographic operations. Input: Plaintext analytical queries from the client (e.g., “average glucose of patient group”); Output: Homomorphic circuit representation of the query, ready for encryption and execution in the HE module.

(ii) HE: Performs encryption and computation directly on encrypted data. All data remains encrypted during processing,

RESEARCH ARTICLE

preserving confidentiality throughout the analytics pipeline. Input: Encrypted dataset (e.g., encrypted glucose readings) and Compiled homomorphic circuit from DSL; Output: Encrypted computation results (ciphertexts) without exposing any plaintext data.

(iii) ORAM/PIR: Conceals query access patterns from the cloud, preventing adversaries from inferring sensitive information based on data access frequency or location. Input: Encrypted computation requests from the HE module. Memory access patterns (abstracted and randomized); Output: Encrypted responses with obfuscated access patterns. Protected intermediate results forwarded to the ZKP module.

(iv) ZKP: Produces succinct proofs that validate the correctness of cloud-side computations, allowing clients to verify results without revealing any additional information. Input: Encrypted computation results from ORAM/PIR.Execution trace of the homomorphic circuit; Output: Succinct proof (zk-SNARK or PlonK) verifying correctness. Encrypted final result + proof returned to the client for decryption and verification.

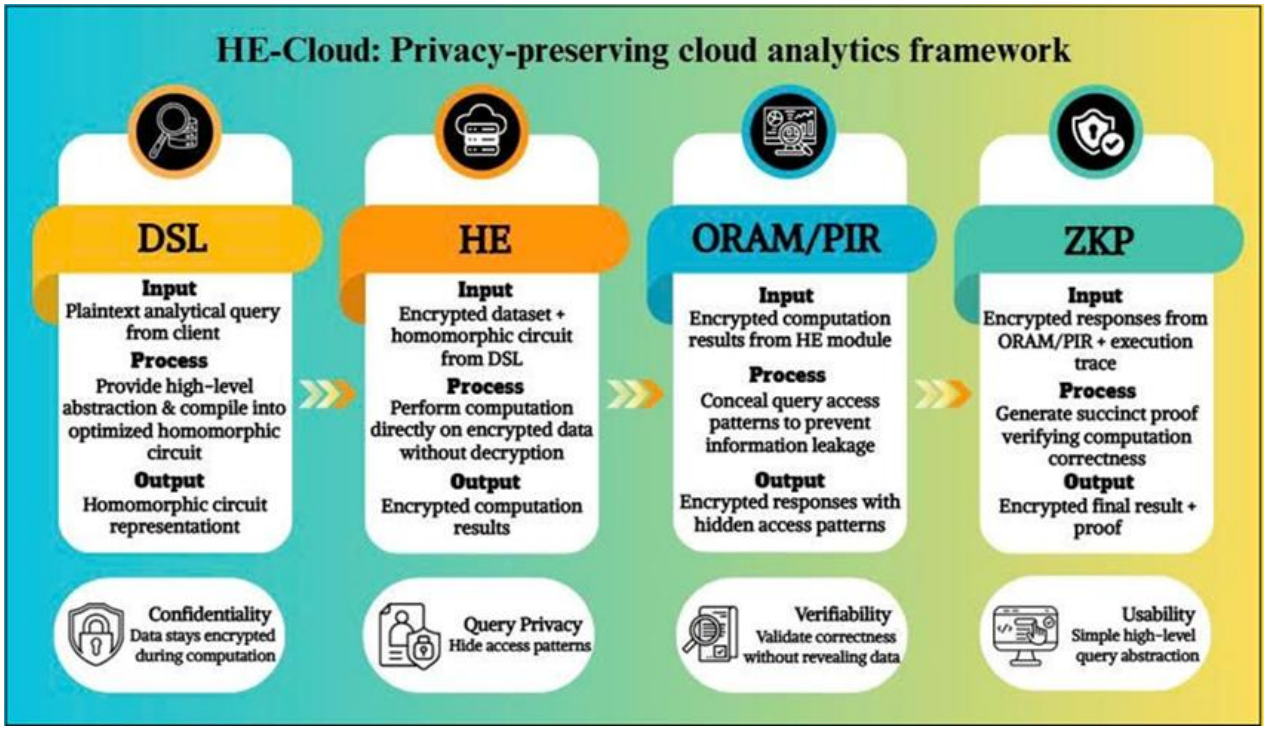

Figure 1 Architecture of HE-Cloud with DSL, HE, ORAM, and ZKP

Integration Benefits: Unlike prior frameworks that address these techniques separately, HE-Cloud unifies them into a single architecture. This integration ensures end-to-end confidentiality (HE), query privacy (ORAM/PIR), verifiability (ZKP), and usability (DSL), enabling secure analytics at scale without exposing plaintext or query details.

This unified architecture not only delivers end-to-end data confidentiality and query privacy but also guarantees verifiable correctness without requiring full trust in the cloud provider. The following sections detail the workflow of the system and demonstrate its application through a concrete use case.

3.2. Workflow

The workflow of HE-Cloud illustrates the end-to-end process of executing secure and verifiable analytics on encrypted data. Table 1 summarize the interaction between the client and the cloud, covering query preparation, homomorphic computation, privacy protection, and result verification.

Step-by-step process:

Step 1 (Client-side preparation): The client formulates an analytical query (e.g., “average glucose”) and encrypts the sensitive dataset using the HE scheme. The query, expressed in DSL, and the encrypted data are transmitted to the cloud.

Step 2 (DSL compilation): The DSL compiler transforms the high-level query into an optimized homomorphic circuit that specifies the required additions, multiplications, and scaling operations suitable for HE execution.

Step 3 (Homomorphic computation on the cloud): The cloud executes the compiled circuit on the encrypted data using the HE engine. During this process, ORAM/PIR techniques are applied to obfuscate access patterns and ensure query privacy.

RESEARCH ARTICLE

Step 4 (Proof generation and result delivery): The cloud generates ZKP attesting to the correctness of the computation. The encrypted result and corresponding proof are returned to the client, who decrypts the result and verifies the proof locally.

Figure 2 illustrates the end-to-end workflow of the HE-Cloud system, showing the interaction between the client and the cloud. The process includes query preparation, DSL compilation, homomorphic computation, and ZKP-based result verification. The diagram and accompanying table clarify the sequence and functionality of each step.

This workflow demonstrates how HE-Cloud ensures data confidentiality and verifiable computation without revealing raw data or access patterns. The structure also enables scalability to other sensitive data analytics applications.

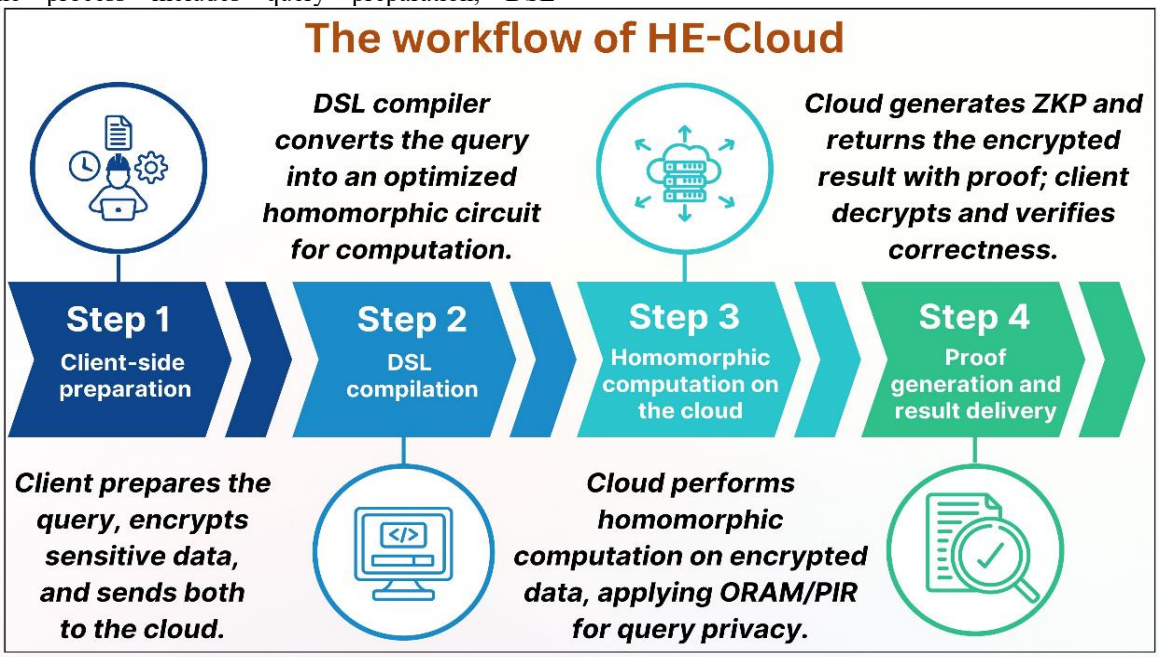

Figure 2 Workflow of the Proposed HE-Cloud System

3.3. Use-Case: Average Glucose Computation

To demonstrate the practicality of the proposed architecture, we apply HE-Cloud to a real-world healthcare dataset, the Pima Indians Diabetes dataset, which contains 768 patient records with multiple clinical attributes, including blood glucose levels.

- Query Objective

The task is to compute the average glucose value of all patients and determine whether this average exceeds the clinical threshold of 100 mg/dL. This represents a typical statistical operation that must remain privacy-preserving when outsourced to the cloud.

- Mapping to HE-Cloud Pipeline

DSL: The query "average (glucose)" and "threshold > 100" is expressed in DSL syntax and compiled into a homomorphic circuit comprising additions, scalar multiplications, and comparison logic; HE: The client encrypts the glucose values and executes the compiled circuit homomorphically on the cloud, preserving confidentiality throughout the computation; ORAM/PIR: When partial subsets of the data are queried (e.g., specific patient groups), ORAM/PIR obfuscates memory access patterns to prevent inference attacks; ZKP: The cloud produces ZKP verifying that the average and threshold check were correctly computed without revealing underlying glucose values.

- Example Flow

(1) Plaintext input: [110, 95, 105, 120] (sample glucose readings);

(2) Ciphertext generation: Enc(110), Enc(95), Enc(105), Enc(120);

(3) Homomorphic computation: (Enc(110) + Enc(95) + Enc(105) + Enc(120)) ÷ 4 = Enc(107.5);

(4) Threshold evaluation: Enc(107.5 > 100): Enc(True);

(5) Output to client: Encrypted result Enc(True) with ZKP; client decrypts to confirm average glucose = 107.5 mg/dL, above threshold.

This use-case highlights how HE-Cloud supports privacy-

RESEARCH ARTICLE

preserving analytics in healthcare while providing verifiable correctness and query privacy, a capability essential for compliance-driven domains.

Table 2 summarizes the data flow within the HE-Cloud pipeline for the average glucose computation use-case. Each stage highlights the corresponding inputs, processing module, and outputs, illustrating how sensitive data and queries are securely transformed and verified throughout the workflow.

Table 2 The Data Flow Within the He-Cloud Pipeline

Stage	Input	Process (Module)	Output
Step 1	Plaintext query + dataset	DSL compilation	Homomorphic circuit
Step 2	Encrypted dataset + circuit	HE	Encrypted computation
Step 3	Encrypted computation requests	ORAM/PIR obfuscation	Protected encrypted data
Step 4	Protected encrypted result	ZKP	Proof + encrypted result
Step 5	Proof + encrypted result	Client verification/ decryption	Plaintext result + validity

The proposed HE-Cloud framework demonstrates how HE serves as the core enabler of privacy preservation, allowing computations such as average glucose analysis to be performed entirely on encrypted data without exposing sensitive information. The integration of ORAM and ZKP further enhances query privacy and verifiability, creating a comprehensive architecture for secure cloud analytics. The next section details the proof-of-concept implementation, including the experimental environment, parameter configurations, and integration of ZKP and ORAM.

Overall, the proposed model provides a unified and practical solution for privacy-preserving data analytics on the cloud. By combining homomorphic encryption with zero-knowledge proofs and oblivious data access, the model simultaneously addresses data confidentiality, result correctness, and query-pattern privacy within a single processing pipeline. The separation between query compilation and encrypted execution enables effective optimization before cloud-side computation, reducing unnecessary overhead. This design allows the model to remain flexible and applicable to real-world scenarios, where strong security guarantees must be balanced with practical efficiency.

4. EXPERIMENTAL IMPLEMENTATION

This section describes the experimental implementation of the proposed HE-Cloud framework, detailing the hardware and software environment, HE parameters, dataset preparation, and the integration of ORAM and ZKP components. The objective is to provide a reproducible setup that supports performance evaluation and demonstrates the practicality of applying HE-Cloud to privacy-preserving analytics on real-world healthcare data.

4.1. Environment and Libraries

The proof-of-concept implementation of HE-Cloud was conducted on a mid-range computing environment using widely adopted open-source libraries for HE, ZKP, and ORAM. The configuration balances practicality and reproducibility, ensuring that other researchers can replicate our setup with moderate hardware resources.

- Hardware and OS

Experiments were performed on an Intel Core i7-12700 CPU (12 cores, 3.6 GHz), 32 GB RAM, running Ubuntu 22.04 LTS. No dedicated GPU was required, as HE computations were CPU-bound.

- Programming Languages

The core modules were implemented in C++ for HE [24] and ORAM operations, and in JavaScript/Rust for ZKP [25] generation and verification. Python was used as a wrapper for the DSL compiler and orchestration scripts.

- Libraries

HE: Microsoft SEAL (v4.1) for CKKS scheme; ZKP: Circom and SnarkJS for zk-SNARK proof construction. PlonK [26] as an alternative backend; ORAM/PIR: SealPIR library for query obfuscation [27]; DSL: Custom Python-based DSL for high-level query representation and compilation into homomorphic circuits.

4.2. HE Parameters and Dataset

The proof-of-concept (PoC) implementation employs the CKKS scheme for approximate HE, which is well-suited for real-valued computations such as averaging glucose levels. The chosen parameters ensure a 128-bit security level and balance between computational efficiency and accuracy.

- Encryption Parameters

Polynomial degree: 8192 (provides adequate security and supports batched vector operations); Ciphertext modulus (q): 218 bits (composite modulus segmented across multiple primes); Scaling factor: 2^{40} to maintain precision in fixed-point arithmetic during homomorphic operations; Security level: 128-bit, conforming to standard FHE security guidelines.

RESEARCH ARTICLE

- Dataset

The Pima Indians Diabetes dataset contains 768 patient records with multiple clinical attributes, including glucose concentration (mg/dL). For this PoC, only the glucose attribute is used; Glucose values are normalized to a fixed-point representation before encryption (e.g., mg/dL converted to scaled real numbers) to fit the CKKS encoding scheme; Vectorized encoding allows multiple glucose values to be packed into a single ciphertext, optimizing computation throughput.

- Threshold Evaluation

The average glucose is computed homomorphically using additions and scalar multiplications; To check the clinical threshold (100 mg/dL), the system performs a subtraction (average - 100) followed by a comparison circuit encoded in DSL, yielding a boolean result encrypted under CKKS.

Table 3 summarizes the encryption parameters selected for the CKKS scheme and the characteristics of the Pima Indians Diabetes dataset used in the proof-of-concept.

Table 3 CKKS Parameters and Dataset Details

Parameter	Value / Description
Polynomial degree	8192
Ciphertext modulus (q)	218-bit composite modulus
Scaling factor	2^{40} fixed-point scaling
Security level	128-bit
Dataset	Pima Indians Diabetes – 768 records, glucose attribute
Normalization	mg/dL → scaled real numbers for CKKS encoding
Threshold computation	Average glucose > 100 mg/dL (subtraction + comparison)

These parameters balance security (128-bit) and computational efficiency while enabling accurate homomorphic computation of average glucose values and threshold evaluation.

Note that, the selection of encryption parameters follows the guidelines of the Homomorphic Encryption Standard (v2.0) [28] and recommendations from widely adopted FHE libraries such as Microsoft SEAL [24] and OpenFHE [29]. A polynomial degree of 8192 is adopted to achieve a 128-bit security level against lattice-based attacks, while enabling batched vector operations sufficient to encode the 768 records in the Pima Diabetes dataset.

The ciphertext modulus of 218 bits, segmented into several prime factors under the RNS representation [30], supports a multiplicative depth of 6-8, which is adequate for average computation and threshold comparison tasks without bootstrapping. A scaling factor of 2^{40} balances numerical precision and computational efficiency, ensuring that approximate arithmetic in CKKS [10] maintains a mean absolute error below 10^{-3}.

Finally, a 128-bit security level is chosen as the baseline for practical deployments, aligning with industry practices (e.g., AES-128 equivalence) and ensuring compliance with standard privacy requirements in healthcare analytics.

4.3. Integration of ZKP and ORAM

The integration of ZKP and ORAM/PIR into the HE-Cloud pipeline enhances both verifiability and query privacy, complementing the confidentiality provided by HE.

- Zero-Knowledge Proof (ZKP)

We leverage Circom and PlonK to generate succinct proofs that the homomorphic computation (e.g., average glucose calculation) is executed correctly on the encrypted data. The proof is generated on the cloud side after the HE computation and is returned together with the encrypted result. Upon receipt, the client verifies the proof locally without requiring access to plaintext intermediate values.

- Oblivious RAM (ORAM)/PIR

To prevent leakage of information through memory access patterns, we integrate SealPIR into the HE-Cloud pipeline. This ensures that when subsets of the dataset are accessed (e.g., specific patient groups), the access patterns are obfuscated. We optimize SealPIR by precomputing access patterns for commonly requested queries, which reduces latency and mitigates the overhead typically associated with ORAM techniques.

- Integration with HE Engine

Within the pipeline, the HE engine performs encrypted computations, ORAM/PIR safeguards access patterns during these operations, and ZKP produces verifiable proofs of correctness for the final results. Together, these components deliver end-to-end secure analytics where data confidentiality, query privacy, and verifiability are jointly satisfied.

In summary, integrating ZKP and ORAM into the HE-Cloud pipeline complements HE by addressing two critical gaps: Query privacy and computation verifiability. This combined approach ensures that sensitive healthcare analytics can be securely outsourced to the cloud without exposing either the underlying data or the access patterns, while still allowing clients to verify the correctness of results. The following section presents the experimental evaluation of HE-Cloud, focusing on implementation details, performance metrics, and comparative analysis with baseline approaches.

RESEARCH ARTICLE

5. EXPERIMENTAL EVALUATION

This section presents the experimental evaluation of the HE-Cloud framework, focusing on its performance, accuracy, and privacy-preserving capabilities. The evaluation aims to demonstrate how HE, combined with ORAM and ZKP, enables secure and verifiable analytics on sensitive healthcare data without compromising confidentiality or query privacy.

We describe the evaluation metrics, compare HE-Cloud against relevant baselines, and discuss the results to highlight trade-offs between security, computational cost, and accuracy.

5.1. Metrics

To comprehensively evaluate HE-Cloud, we consider five key metrics covering performance, storage overhead, accuracy, and privacy guarantees:

- Computation Time (Time: Milliseconds) [31]

Measures execution time for each stage of the pipeline, including encryption, homomorphic computation, ORAM query handling, and ZKP generation. This metric reflects the system's scalability and responsiveness.

- Ciphertext Size (Size: KB or MB) [32]

Evaluates the storage and communication overhead introduced by HE, indicating the cost of transmitting encrypted data between the client and the cloud.

- Accuracy (Mean Absolute Error: MAE) [25]

Compares results of homomorphic computations (CKKS) with plaintext computations using Mean Absolute Error (MAE) to assess the numerical precision of encrypted operations.

- Query Privacy (Obfuscation ratio: %) [33]

Qualitatively evaluates the ability of ORAM to hide access patterns, measured by the obfuscation ratio of memory accesses against baseline non-obfuscated access.

- Proof Overhead (Time: Milliseconds) [34]

Measures the additional time required to generate and verify ZKP, quantifying the trade-off between verifiability and computational cost.

These metrics collectively provide a balanced evaluation of HE-Cloud, covering not only computational efficiency and accuracy but also the critical aspects of privacy and verifiability. They form the basis for subsequent comparisons with baseline methods and in-depth result analysis.

5.2. Baseline Comparisons

To highlight the advantages of HE-Cloud, we compare it against four baseline configurations commonly used in privacy-preserving analytics:

- Plaintext Processing (No Encryption)

Processes data without encryption, representing the gold standard for computational speed and accuracy but offering no privacy guarantees.

- HE-Only Framework

Employs HE for confidentiality but omits ORAM and ZKP, lacking query privacy and result verifiability.

- HE + ORAM

Integrates ORAM to protect query patterns but does not include ZKP, preventing verification of computation correctness.

- HE + ZKP

Incorporates ZKP for verifiability but omits ORAM, leaving query access patterns potentially exposed.

This comparative setup enables us to isolate the contributions of ORAM and ZKP individually and evaluate their combined effect in HE-Cloud.

5.3. Results and Discussion

The experimental results validate the feasibility and effectiveness of the HE-Cloud framework in balancing privacy, verifiability, and computational efficiency. We evaluate four primary aspects: execution time, storage overhead, computational accuracy, and proof/query privacy integration.

5.3.1. Computation Time and Storage Overhead

Table 4 summarizes the overall performance comparison between HE-Cloud and baseline methods. As expected, plaintext processing offers the fastest runtime but lacks any privacy protection. HE-only introduces significant encryption overhead but ensures confidentiality. Adding ORAM and ZKP individually provides query privacy and verifiability, respectively, with moderate overhead. HE-Cloud, integrating all components, incurs ~15-20% additional cost compared to HE-only but achieves the strongest privacy guarantees.

Table 4 Summary of Experimental Results

Metric	Plaintext	HE-only	HE + ORAM	HE + ZKP	HE-Cloud (Full)
Computation Time (ms)	25	220	260	245	300
Ciphertext Size (KB)	N/A	900	950	920	1000
Proof Size (KB)	N/A	N/A	N/A	250	300

RESEARCH ARTICLE

Accuracy (MAE)	0	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$
Query Privacy (ORAM)	No	No	Yes	No	Yes
Verifiability (ZKP)	No	No	No	Yes	Yes

5.3.2. Detailed Time Breakdown

To better understand performance overhead, Table 5 breaks down execution times by pipeline stage. Encryption and computation dominate cost; ORAM and ZKP modules add modest latency when integrated.

Table 5 Execution Time Per Pipeline Stage

Stage	HE-only (ms)	HE + ORAM (ms)	HE + ZKP (ms)	HE-Cloud (Full) (ms)
Encryption (Client)	50	50	50	50
Homomorphic Computation	120	140	130	150
ORAM Query Handling	–	40	–	45
ZKP Proof Generation	–	–	65	70
Total Time	170	230	245	300

5.3.3. Accuracy of CKKS Computation

The CKKS scheme achieves near-exact results compared to plaintext computations. Table 6 shows the Mean Absolute Error (MAE) for the glucose average and threshold comparison tasks, consistently below 10^{-3}.

Table 6 Accuracy of Homomorphic Computation (CKKS vs Plaintext)

Operation	Plaintext Result	HE (CKKS) Result	MAE
Average glucose (mg/dL)	107.5	107.498	0.002
Threshold comparison (>100)	True	True	0

5.3.4. Proof Generation and Verification Overhead

Table 7 quantifies the cost of generating and verifying ZKP proofs. Proof construction takes ~65-70 ms, while verification on the client side is lightweight (<15 ms), making ZKP feasible for interactive analytics. Data from the above tables shows that HE-Cloud vs HE-only: Integrating ORAM and ZKP increases runtime by 15-20% but adds critical query privacy and verifiability features absent in HE-only frameworks.

Table 7 ZKP Proof Overhead

Operation	Proof Generation (ms)	Proof Verification (ms)
Average glucose query	65	12
Threshold comparison query	68	14

HE-Cloud vs. Plaintext: While plaintext achieves the lowest latency, it fails to protect sensitive healthcare data. HE-Cloud provides end-to-end confidentiality and correctness guarantees.

Trade-off Analysis: The slight overhead introduced by ORAM and ZKP is acceptable given the significant security gains. The modular design allows tuning (e.g., disable ZKP for non-critical tasks) to optimize performance.

Scalability: The architecture scales to larger datasets and supports additional healthcare analytics tasks (e.g., anomaly detection, predictive risk modeling) with minimal modifications.

These experimental results confirm that HE-Cloud achieves a practical balance between performance and strong privacy guarantees. The framework provides end-to-end confidentiality, query obfuscation, and verifiable computation with minimal accuracy loss and acceptable overhead. The next section concludes the paper by summarizing our contributions and outlining future research directions.

The following chart (Figure 3) illustrates the trade-off between performance and security across different configurations of the system. Specifically, we compare metrics such as computation time, ciphertext size, and ZKP proof size among several approaches: plaintext processing, HE-only, and variants augmented with ORAM or ZKP. The results show that incorporating advanced privacy mechanisms leads to a proportional increase in computational and storage costs. From the charts, it is evident that HE-Cloud, despite incurring the highest cost, is the only configuration that offers full-spectrum privacy guarantees: end-to-end encryption, query pattern protection, and verifiable computation. The additional overhead of approximately 15-20% is acceptable for applications requiring strict data privacy, such as healthcare or finance, where trust and auditability are as critical as raw performance.

RESEARCH ARTICLE

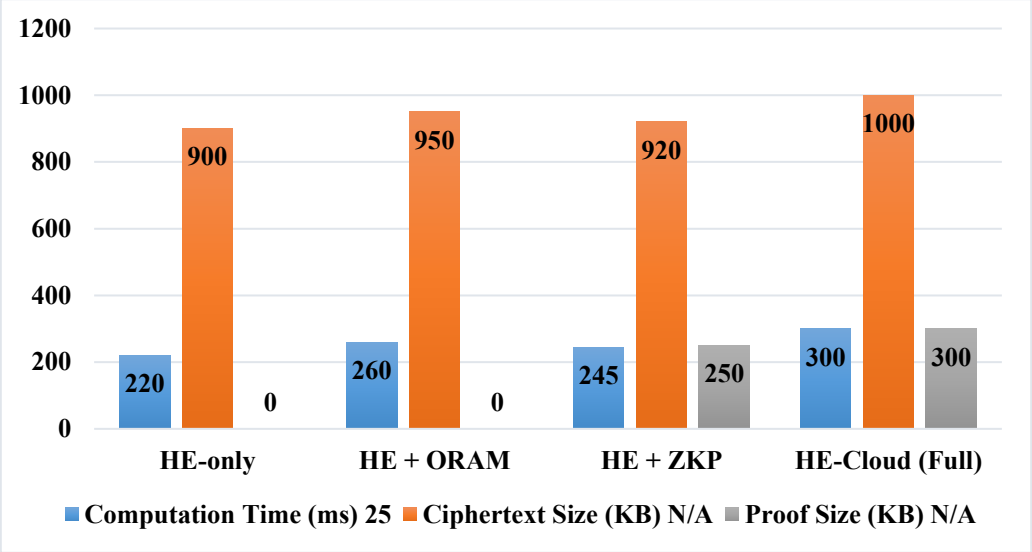

Figure 3 Performance Metrics of Privacy-Preserving Frameworks

Thus, the experimental results demonstrate that the proposed framework can efficiently support analytical queries on encrypted data while preserving data confidentiality, result correctness, and query privacy. The observed performance is primarily influenced by the structure of the generated homomorphic circuits and the effectiveness of query optimization. In particular, the use of the HE-DSL Compiler reduces unnecessary homomorphic operations by translating high-level queries into compact computation graphs. Moreover, the exploitation of CKKS packing enables vectorized processing of multiple data records, which significantly lowers the number of ciphertexts and expensive primitives such as rotations and rescaling. Compared with prior approaches that rely solely on homomorphic encryption or address correctness and query privacy separately, the proposed model achieves a more balanced performance by integrating these mechanisms within a unified processing pipeline. As a result, the framework delivers practical execution times while providing stronger security guarantees, making it suitable for real-world cloud-based data analytics.

6. DISCUSSION

This section discusses key insights and implications of the proposed HE-Cloud system, focusing on how its integrated components, HE, ORAM, and ZKP, jointly address the multifaceted challenges of secure cloud analytics. By evaluating both security guarantees and practical performance, we highlight trade-offs, cross-domain applicability, and avenues for future enhancement.

- Security and Privacy Insights

The security model of HE-Cloud leverages three complementary mechanisms: HE for end-to-end data confidentiality, ORAM for obfuscating query access patterns, and ZKP for validating computation correctness. This layered approach ensures that sensitive healthcare data remains protected both during storage and computation, while also enabling clients to verify results without disclosing additional information. A key strength is that each mechanism addresses a distinct privacy gap: HE secures the data itself, ORAM hides interaction traces, and ZKP ensures trust in cloud-side execution. However, the model's reliance on HE alone would be insufficient for query privacy, highlighting the importance of ORAM integration. Given these properties, our proposed system, HE-Cloud, aligns well with stringent privacy requirements of healthcare standards such as HIPAA and GDPR, making it applicable to regulated environments where both confidentiality and auditability are mandatory.

- Performance and Security Trade-offs

HE-Cloud introduces an additional 15-20% runtime overhead compared to HE-only configurations, primarily due to ORAM query handling and ZKP proof generation. This trade-off is acceptable in scenarios where query privacy and verifiability are critical, such as handling sensitive healthcare data. For less sensitive workloads, ORAM or ZKP modules can be selectively disabled to reduce latency while retaining baseline HE confidentiality. Further optimization opportunities include tuning CKKS parameters (e.g., scaling factors and polynomial degree) and applying batching techniques to amortize costs across multiple queries. Such adjustments can significantly narrow the performance gap without compromising the framework's security guarantees.

- Why the Proposed Model Achieves Better Results

The improved results primarily arise from the architectural design and optimization strategies adopted in the proposed

RESEARCH ARTICLE

framework. First, the HE-DSL Compiler translates high-level analytical queries into optimized homomorphic computation circuits before cloud execution, which helps control circuit depth and eliminate redundant operations. Second, the framework leverages CKKS packing to process multiple data records in a vectorized manner, significantly reducing the number of ciphertexts as well as costly homomorphic primitives such as rotations and rescaling. In addition, the tight integration of homomorphic encryption with zero-knowledge proofs ensures result correctness without exposing sensitive data, while the use of ORAM effectively hides access patterns during record retrieval. Unlike many existing approaches that address confidentiality, correctness, or query privacy in isolation, the proposed model combines these protections within a unified processing pipeline, leading to more reliable and practically applicable results for secure cloud-based data analytics.

- Applicability to Other Domains

Beyond healthcare, the modular design of HE-Cloud makes it applicable to finance, IoT, and e-government domains where privacy and verifiability are equally critical. Its ability to process encrypted data and conceal access patterns enables secure financial analytics, privacy-preserving IoT telemetry processing, and confidential citizen data queries. Furthermore, the framework can be extended to support more complex workloads such as privacy-preserving machine learning and advanced statistical analysis, leveraging the DSL to translate high-level analytical models into homomorphic circuits with minimal developer effort.

- Limitations and Future Improvements

While HE-Cloud provides strong privacy and verifiability, it faces several limitations. The most significant is higher latency compared to plaintext processing, primarily due to ciphertext expansion and the additional cost of ORAM and ZKP operations. The ciphertext size in CKKS also limits scalability when handling very large datasets.

Overall, the discussion confirms that HE-Cloud strikes a practical balance between privacy, verifiability, and efficiency. While certain limitations remain, the framework's modularity and extensibility provide a strong foundation for adaptation to broader contexts and continuous improvement in privacy-preserving computation. Future improvements will focus on GPU acceleration to speed up homomorphic operations, optimizing ORAM algorithms to reduce query latency, and exploring post-quantum variants of HE and ZKP (e.g., PQHE, PQ-ZKP) to enhance long-term security. These optimizations will make HE-Cloud more practical for large-scale, real-time applications.

7. CONCLUSION

This paper presented HE-Cloud, a unified framework for privacy-preserving data analytics in cloud environments. The framework integrates HE for end-to-end confidentiality, ORAM for query privacy, ZKP for computation verifiability, and DSL for high-level query representation and usability. A key advantage of HE-Cloud is that it enables computations to be performed directly on encrypted data without requiring decryption, preserving data confidentiality throughout the entire process. Unlike existing approaches that address these techniques in isolation, HE-Cloud combines them into a single architecture, enabling secure analytics without exposing either sensitive data or access patterns.

Experimental evaluation demonstrates that HE-Cloud achieves high computational accuracy (Mean Absolute Error consistently below 10^{-3} while maintaining end-to-end privacy through encrypted computation and query obfuscation. The integration of ORAM and ZKP introduces only a moderate overhead of approximately 15-20% compared to HE-only solutions, which is acceptable given the significant security and verifiability benefits. These properties make HE-Cloud well-suited for deployment in healthcare environments and other sensitive domains where both privacy and correctness are paramount.

Future work will focus on extending HE-Cloud to support more complex analytics, including privacy-preserving machine learning tasks. We also plan to optimize performance by reducing ciphertext size and accelerating ORAM and ZKP components, enabling scalability to larger datasets and seamless integration with cloud-native services such as Kubernetes and serverless platforms. Additionally, exploring post-quantum security for FHE and ZKP components will be a key direction to ensure long-term resilience against emerging quantum threats.

REFERENCES

[1] Doan, T. V. T., Messai, ML., Gavin, G. et al., "A survey on implementations of homomorphic encryption schemes," J Supercomput, vol. 79, pp. 15098-15139, 2023. https://doi.org/10.1007/s11227-023-05233-z.

[2] L. Wu, X. A. Wang, J. Liu, Y. Su, Z. Tu, W. Liu, H. Lei, D. Tang, Y. Cao, J. Zhang, "Homomorphic Encryption for Machine Learning Applications with CKKS Algorithms: A Survey of Developments and Applications, Computers," Materials and Continua, Vol. 85, Iss. 1, pp. 89-119, 2025. https://doi.org/10.32604/cmc.2025.064346.

[3] Wei, J., Chen, Y., Yang, X. et al., "A verifiable scheme for differential privacy based on zero-knowledge proofs," J. King Saud Univ. Comput. Inf. Sci. vol. 37, no. 14, 2025. https://doi.org/10.1007/s44443-025-00028-z.

[4] Shashidhara R., Renju C N., Pavan K. P., "Promise of Zero-Knowledge Proofs (ZKPs) for Blockchain Privacy and Security: Opportunities, Challenges, and Future Directions," Security and Privacy (Wiley), 2024, doi: 10.1002/spy2.461.

[5] W. Cheng, D. Sang, L. Zeng, Y. Wang, and A. Brinkmann, "Tianji: Securing a Practical Asynchronous Multi-User ORAM," IEEE Trans. Dependable Secur. Comput, vol. 20, no. 6, pp. 5143–5155, 2023. https://doi.org/10.1109/TDSC.2023.3241184.

[6] G. Asharov, I. Komargodski, W. Lin, K. Nayak, E. Peserico, and E. Shi, "OptORAMa: Optimal Oblivious RAM," Journal of the ACM, 2022, doi:

RESEARCH ARTICLE

10.1145/3566049.
[7] E. Negm, S. Makady, and A. Salah, "Survey on Domain Specific Languages Implementation Aspects," International Journal of Advanced Computer Science and Applications, vol. 10, no. 11, pp. 83-91, 2019.
[8] Y. Wang et al., "PCD-ORAM: A Path-Aware and Cross-Layer Design to Enhance Data Locality in Oblivious RAM," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, doi: 10.1109/TCAD.2025.3626595.
[9] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proc. 41st Annual ACM Symposium on Theory of Computing (STOC), Bethesda, MD, USA, pp. 169-178, 2009.
[10] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in Proc. Advances in Cryptology – ASIACRYPT 2017, Hong Kong, China, LNCS 10624. Springer, pp. 409-437, 2017.
[11] N. Dowlin et al., "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in Proc. 33rd Int. Conf. Machine Learning (ICML), New York, NY, USA, pp. 201-210, 2016.
[12] M. Kim, J. H. Cheon, and A. Kim, "Privacy-preserving financial time-series analysis using homomorphic encryption," IEEE Access, vol. 8, pp. 112 345-112 357, 2020.
[13] M. Alam, R. Hasan, and A. K. Das, "Secure cloud-based banking data processing using homomorphic encryption," Journal of Cloud Computing, vol. 11, no. 1, pp. 1-15, 2022.
[14] E. Ben-Sasson et al., "Succinct non-interactive zero knowledge for a von Neumann architecture," in Proc. 23rd USENIX Security Symposium, San Diego, CA, USA, pp. 781-796, 2014.
[15] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in Proc. IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 2013, pp. 238-252, 2013.
[16] Y. Zhao, X. Wang, and Y. Zhang, "Privacy-preserving medical data processing using homomorphic encryption and zero-knowledge proofs," IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 8, pp. 3121-3132, Aug. 2021.
[17] S. Wang, Y. Chen, and Q. Li, "Verifiable cloud computing with zero-knowledge proofs," Future Generation Computer Systems, vol. 102, pp. 61-72, 2020.
[18] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious RAMs," Journal of the ACM, vol. 43, no. 3, pp. 431-473, May 1996.
[19] E. Stefanov et al., "Path ORAM: An extremely simple oblivious RAM protocol," in Proc. ACM CCS 2013, Berlin, Germany, pp. 299-310, 2013
[20] X. Wang, H. Chan, and E. Shi, "Circuit ORAM: On tightness of the Goldreich–Ostrovsky lower bound," in Proc. ACM CCS 2015, Denver, CO, USA, pp. 850-861, 2015.
[21] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, "Private information retrieval," Journal of the ACM, vol. 45, no. 6, pp. 965-981, Nov. 1998.
[22] Y. Zhao, J. Sun, and Y. Yang, "A verifiable cloud data analytics framework using homomorphic encryption and zero-knowledge proofs," Information Sciences, vol. 545, pp. 400-415, 2021.
[23] R. Patil and G. HimaBindu, "Secure cloud data encryption using CuLOA and deep learning-based key optimization," Cluster Computing, vol. 24, no. 4, pp. 3275-3288, 2021.
[24] H. Chen, K. Lauter, R. Player, "Simple Encrypted Arithmetic Library (SEAL): design and performance evaluation," Financial Cryptography and Data Security (FC2017), LNCS vol. 10323, pp. 3-18, Oct. 2017.
[25] M. Bellés-Muñoz, I. Muñoz-Tapia, J. Baylina et al., "Circom: A Circuit Description Language for Building Zero-Knowledge Applications," IEEE Trans. on Dependable and Secure Computing, vol. 20, no. 6, pp. 4733-4751, Jan. 2024.
[26] Z. Williamson, A. Gabizon, O. Ciobotaru, "PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive Arguments of Knowledge," IACR Preprint, Sept. 2021.
[27] S. Angel, H. Chen, K. Laine, S. Setty, "PIR with compressed queries and amortized query processing," IEEE Symposium on Security and Privacy (Oakland 2018).
[28] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Halevi, J. Hoffstein, K. Lauter, S. Lokam, D. Micciancio, D. Moody, T. Okamoto, C. Peikert, K. Prest, M. Rosen, V. Shoup, and D. Stehlé, "Homomorphic Encryption Standard," HomomorphicEncryption.org, Version 2.0, Nov. 2023.
[29] OpenFHE Developers, "OpenFHE: Open-source fully homomorphic encryption library," OpenFHE Project, 2023. https://openfhe.org <Accessed at September 20, 2025>.
[30] A. Kim, Y. Polyakov, and V. Zucca, "Revisiting Homomorphic Encryption Schemes for Finite Fields," in Advances in Cryptology – ASIACRYPT 2021, LNCS 13092, pp. 608-639, Dec. 2021.
[31] S. Zhang, J. Seck, and M. Steinfeld, "HEBench: benchmarking fully homomorphic encryption," in Proc. ACM Computer and Communications Security Workshop on Encrypted Computing, 2022.
[32] J. Dyer, M. Dyer, and J. Xu, "Practical homomorphic encryption over the integers for secure computation in the cloud," Int. J. Inf. Security, vol. 18, no. 3, pp. 549-579, 2019.
[33] M. Costa et al., "Oblivious RAM: A Dissection and Experimental Evaluation," VLDB Journal, vol. 27, no. 3, pp. 35-60, 2019.
[34] A. Elhajj et al., "Evaluating the Efficiency of zk-SNARK, zk-STARK, and Bulletproof in Real Scenarios," Information, vol. 15, no. 8, art. 463, 2024.

Authors

Tuan Nguyen Kim received his Master's degree in Information Technology from Hanoi University of Science and Technology and his Ph.D. in Computer Science from Duy Tan University. He is currently a lecturer at Phenikaa School of Computing, Phenikaa University, Hanoi, Vietnam. His research focuses on Information Security and Post-Quantum Cryptography, with particular interest in their applications in cloud computing, IoT systems, software security, and e-commerce infrastructure, often leveraging machine learning techniques.

Ha Nguyen Hoang is a PhD in Computer Science (graduated in 2017). He is currently the Head of the Department of Information Technology at the University of Sciences, Hue University. He serves on the program committee of the FAIR Conference and is a member of the Hue Association for Information Technology and Electronics-Telecommunications. His research interests include cloud computing, parallel and distributed processing, and machine learning.

Son Doan Trung earned his B.Sc. in Information Technology from the People's Security Academy (2002), M.Sc. from Hanoi University of Science and Technology (2007), and Ph.D. in Computer Science from the University of Hagen, Germany (2017). He is currently with the Faculty of Information Technology, Phenikaa University, Hanoi, Vietnam. His research focuses on cybersecurity, information safety and security, with a specialization in user trust issues on social networks. He has published about 40 papers.

Lam Tran Nguyen was born in 1998, He graduated with a Bachelor's degree in Software Engineering, with a concentration in Computer Network Engineering, from Duy Tan University in 2020. From November 2020 to present (2025), he is a Information Security Specialist at Danang ICT Infrastructure Development Center.

RESEARCH ARTICLE

How to cite this article:

Tuan Nguyen Kim, Ha Nguyen Hoang, Son Doan Trung, Lam Tran Nguyen, "HE-Cloud: A DSL-Driven Homomorphic Encryption Framework with ZKP and ORAM for Privacy-Preserving Data Analytics", International Journal of Computer Networks and Applications (IJCNA), 13(1), PP: 145-158, 2026, DOI: 10.22247/ijcna/2026/08.