(Online) ISSN 2712-0562
Original Research

Sustainable Engineering and Innovation
Vol. 8, No. 1, 2026, pp.219-234
https://doi.org/10.37868/sei.v8i1.id541

Efficient task-verification and data collaboration processing in mobile-cloud based application using ZKP and SMPC

Matheen Fathima G.[1*], Shakkeera L.[2]
[1,2] Presidency School of Computer Science and Engineering, Presidency University, Karnataka, India

*Corresponding author E-mail: matheenfathima08@gmail.com

Received Jul. 10, 2025
Revised Apr. 21, 2026
Accepted May 5, 2026
Online May 20, 2026

Abstract

With the increasing adoption of mobile applications, data in the mobile cloud faces numerous security threats and privacy breaches. To overcome cyberattacks, ensuring confidentiality and data security for users' sensitive data is pivotal in mobile cloud computing. Traditional security mechanisms involve data leakage during the verification process, while blockchain-dependent solutions lead to high resource consumption and latency. Additionally, collaborative data processing during data transactions can result in potential privacy attacks on users.

This paper proposes a novel approach for maintaining a security framework for Microservice-based Mobile Cloud Computing (MSCMCC) using hybrid cryptographic frameworks such as Zero-Knowledge Proof (ZKP) and Secure Multi-Party Computation (SMPC). The proposed model validates users' offloaded data using zk-SNARK and Groth16 for task verification and enables data analysis from multiple users without exposing raw data. SMPC is employed for privacy preservation during collaborative multi-party computation. Experimental results demonstrate that the proposed framework reduces power consumption, improves energy efficiency during processing by 30–35%, lowers computational costs, enhances security and privacy, and effectively manages dynamic load balancing compared to traditional cryptographic techniques.

Keywords: mobile cloud computing, Zero-knowledge proof, Secure multi-party computation, task verification, data collaboration, data security and confidentiality.

1. Introduction

The rapid growth of microservice-based mobile cloud computing [1] has transformed the landscape of mobile applications by enabling efficient task offloading and resource allocation across the distributed cloud environments. However, the intrinsic features of MSCMCC introduce critical security and privacy risks, including data exposure, unauthorized access, and threats to computational integrity. Traditional security techniques, such as encryption and authentication, lack the ability to provide end-to-end privacy assurances during task verification and multi-party computations, which remains a significant obstacle for mobile clouds susceptible to adversarial cyberattacks that remain an obstacle.

The existing MSCMCC framework faces challenges related to data leakage during task validation, privacy risks in the collaborative processes [2], and excessive computational resource consumption in cryptographic methods.

When mobile devices offload tasks to cloud servers, sensitive data may become vulnerable, leading to privacy breaches and unauthorized data access. The lack of an efficient decentralized security mechanism increases the risk of malicious computation manipulation and unauthorized execution.

This paper focuses on a novel security framework that combines ZKP [3] and SMPC [4] for MSCMCC. ZKP optimizes task validation using privacy-preserving techniques and authorizes mobile devices without disclosing sensitive data during the computation process. SMPC supports secure data collaboration, ensuring that different cloud service providers perform their computations on encrypted data without accessing raw data. This combined framework optimizes confidentiality, integrity, and trust while reducing the computational overhead and optimizing system efficiency. Figure 1 illustrates how ZKP uses ZK-SNARK to authenticate the computations, ensuring that results are obtained without disclosing the inputs or sensitive data.

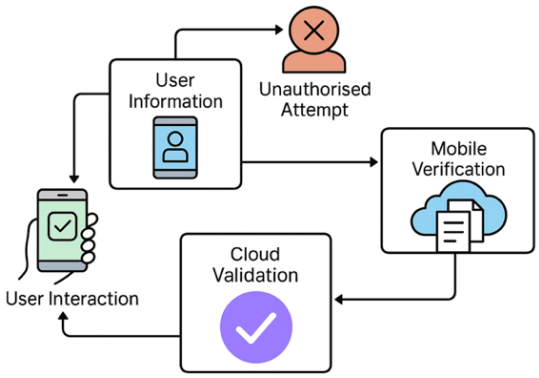

Figure 1. Verification in mobile cloud

SMPC uses the secret Shamir's secret sharing scheme to split data into fragments to distribute across the nodes. When multiple participants join to perform the data collaboration process, they split the data into fragments (i.e., puzzle pieces). The linear operations are performed on the encrypted data, and the non-linear operations use Beaver triples to mask the complex data. The ZKP-SMPC framework reduces the computational overhead, mitigates privacy risks, and further strengthens the security of MSCMCC applications. It eradicates the centralized authorities by incorporating privacy-maintaining cryptography methods for scheduling, execution, and verification of tasks.

Compared to the ECC+ZKP approach, which relies heavily on authentication, cryptographic mechanisms, and blockchain concepts, the proposed approach avoids dependance on decentralized ledgers for transaction verification. This approach relies on a decentralized ledger for verifying the trust on each transaction record. The proposed system verifies the microservice transaction using SMPC and ZKP with no delay of scalability and latency which is created by the distributed application. While traditional approach depends on blockchain technology (ECC+ZKP) and (blockchain-based ZKP), the proposed system provides a blockchain-independent ZKP and SMPC solution for the MSCMCC. This system ensures fine-grained task-level validation with zk-SNARK/ Groth16 for validating each microservice without disclosing input data. It supports privacy-preserving multi-party computations for various cloud servers, while minimizing computational complexity, reducing energy consumption, and requiring less memory space.

The manuscript is structured as follows: Section II describes the related works. Section III presents the proposed hybrid cryptographic framework and outlines the mathematical notation used in this work. Section IV evaluates the performance of the proposed approach in comparison with existing models and analyzes the performance of the simulation environment. Section V concludes the manuscript.

2. Literature review

MCC deals with critical vulnerabilities in data security and integrity; the conventional password-based system battles against emerging cyber risks [5]. This paper combines hybrid models such as ElGamal encryption and MD5 hashing. ElGamal is used for data confidentiality through public-private keys, where MD5 verifies user

authentication. This model achieves memory efficiency of approximately 85-95% and improved speed, optimized for mobile platforms, where ElGamal increases computational cost. This model is less intrusive than the conventional methods, optimizing both security and performance.

Cloud storage [6] faces major security issues; standalone encryption techniques fail to assure confidentiality and integrity against increased risk. Hybrid models such as ECC and ZKP can be combined with these models to encrypt privacy-preserving verification. ECC is used for advanced encryption, while ZKP enables secure verification without disclosing sensitive information. HECCZKP secures data against covert attacks and unauthorized access by linking encrypted storage with ZKP. This assures data protection, proof, integrity and verifiability in untrusted cloud infrastructures. The proposed system balances scalability and security, optimizing it for cloud environments with constrained resources.

The IoMT [7] requires lightweight, secure protocols to protect patient data in a resource-constrained environment. ECC provides optimized encryption with small key sizes compared to RSA, optimizing resource utilization. ZKP-based ECC protocols validate incognito authentication and secure key exchange, reducing the risk of interception attacks. The experimental results show that the 300 ms time taken for executing 160-bit operations exceeds conventional non-interactive ZKP systems. The proposed system ensures undetectable communication while maintaining privacy and efficiency.

Standard cloud key agreements depend on centralized entities, exposing them to critical vulnerabilities and attacks [8]. DKG improves decentralization but suffers from inefficiency in dynamic member handling and fault tolerance. Blockchain ensures authentication but increases response time in large-scale networks. ZKP provides secure, limited-exposure verification yet needs enhancing for low-latency execution. Multi-cloud segment availability, but does not provide cryptographic protection against collusion. The proposed system combines DKG for versatile fault tolerance, zk-SNARK, and quantum-safe protocols for scalable, robust, and secure cloud key management.

Smart city frameworks use IoT sensors and edge computing [9] for real-time data monitoring, which faces scalability and constant delay in dynamic urban environments. MCC enables efficient resource allocation for mobile services, but security risks arise during the data transmission and storage in a centralized network. Biometric verification enhances user identification however, it raises privacy issues due to the risk of template breaches or misuse. AI-powered analytics enhance traffic, energy, and infrastructure management, but the dependence on cloud-based solutions leads to delays and energy inefficiency. Edge computing with ML reduces response time however, it faces the challenge of standardizing various sensor data formats and quality.

The telehealth system depends on standard cryptographic keys, which increases the risks of computational overhead and the breach points to hardware tampering [10]. Traditional methods use authentication like certificates and biometrics, which struggle to monitor the security constraints. The PUF for hardware-based authentication improves device trust but lacks integration in a real-time environment [11]. ZKP is implemented for privacy concerns in distributed healthcare services [12]. The proposed system uses ZKP-MAC with ZKP to authorize lightweight, tamper-resistant verification of patients. This system ensures security and privacy guarantees in a resource-constrained telehealth environment.

The IoMT [13] system faces problems in authentication, which leads to security and efficiency. It solely relies on cryptographic methods, which are vulnerable to threats and collusion. The traditional blockchain-based system leads to latency and scalability issues, while traditional ZKP faces auditability problems in multi-domain healthcare networks. The proposed LSAC system in IoMT by integrating SSI with quantum-resistant ZK-Stark and PLANK. It uses a Hyperledger-based consortium blockchain for trust, decentralized access, scalability, and auditing with less computational overhead in dynamic healthcare environments.

The fog/edge [14] authentication protocol endures from synchronization susceptibility of timestamps depending on hash chains, and also lacks commutable modes for the dynamic network. The centralized distribution of keys

leads to single-point failure, while blockchain integration leads to overhead for latency. The proposed HCFE protocol is a mode-based hash chain with ZKP to authorize quantum-resistant mutual authentication between servers and IoT edge devices in a heterogeneous network. This protocol manages the computational overhead in distributed cloud architectures in a micro-cloud (Table 1).

Table 1. Comparative analysis of cryptographic techniques in cloud and IoT environments

Literature	Pros	Cons
Misra et.al 2025	It uses ECC for encryption in IOMT for security. ZKP is used for anonymous communication and privacy.	Integrating the ZKP protocol and ECC in the IOMT devices results in computational delay, scalability, and complexity.
Aravindhan et. al 2025	In MCC data integrity and authentication always remain an obstacle.MD5+ElGamal encryption provides a higher level of security.	ElGamal encryption works slower, it also creates deduplication and a lack of memory management. MD5 is vulnerable to attacks.
Jansirani et. al 2025	HECCZKP algorithm combines ECC and ZKP to provide data integrity and security in the cloud.	This algorithm faces computational overhead, complexity, and difficulty in scaling the resource in the cloud.
Balaram et.al 2024	Integrating DKG for decentralization and ZKP protocol for verification. To preserve privacy in a distributed cloud environment.	The complexity arises due to the blockchain used for data integrity, which increases latency in the cloud.

3. Proposed system

This system combines ZKP and SMPC-based threshold cryptography to implement end-to-end security, task authentication, and decentralized collaboration. Figure 2 illustrates that the mobile devices initiate the requests through the portal (TLS 1.3 secured API gateway) that directs the queries to the ZKP engine for cryptographic verification.

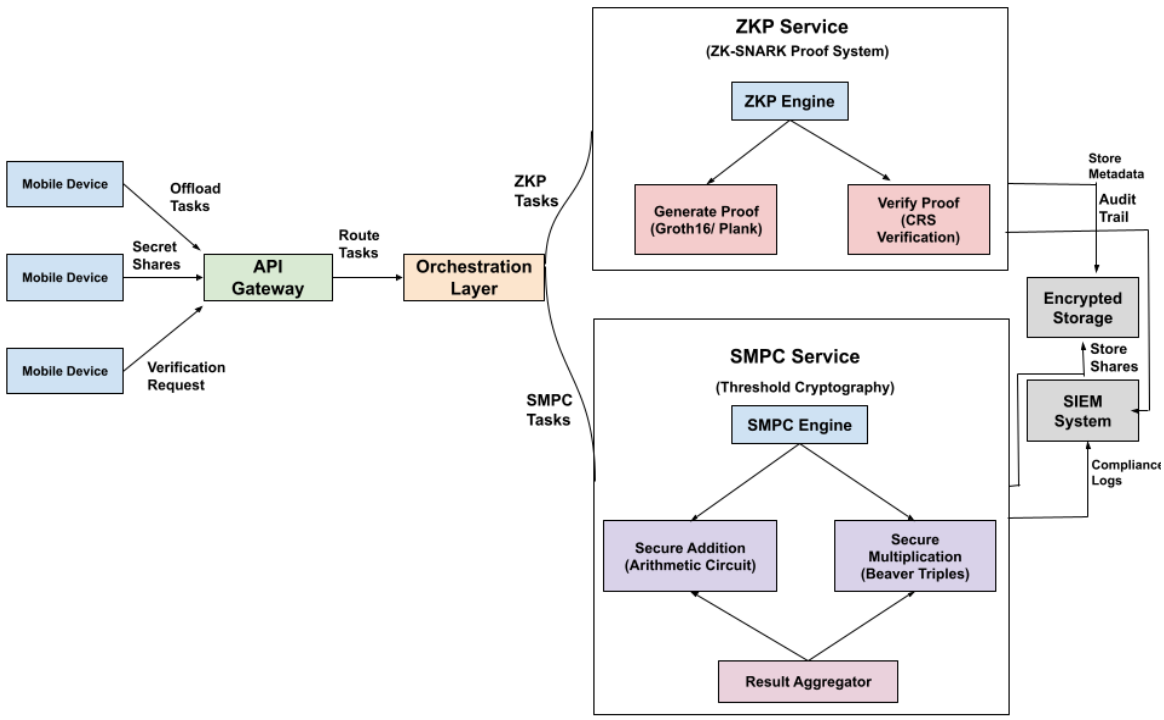

Figure 2. ZKP-SMPC Framework

The Security Information and Event Management (SIEM) acts as a storage module, which uses ZK-SNARK protocols to generate succinct proofs (π) for the mobile tasks, and validates using verify zone using a verification key ($z\kappa$) and public input (p). This ensures that the task is authorized without raw data being exposed. The data at rest is secured through encrypted storage, while the SMPC engine (t, n) splits the sensitive data into n parts, requiring t parties to collaborate for a rebuild. The access layer imposes Role-Based Access Control (RBAC) with OAuth 2.0/mTLS, and security testing on its sub-systems to perform continuous vulnerability scans. The device-appropriate policies validate the device integrity and execute geo-fencing for secure mobile-to-storage workflows. The dataset consists of post-COVID vaccination heart attack data [15], which is trained using a random forest classifier to predict cardiac risk using features like age, BP, cholesterol, BMI, and diabetes status. The mobile device task is offloaded to the cloud for processing using a hybrid ZKP-SMPC framework. Each offloaded task is verified using the ZKP module. The task verification process is modeled as a function $V(t_i, \pi_i) \rightarrow \{0,1\}$, where t_i is an offloaded task and π_i is a zk-SNARK proof validated using the Groth16 pairing equations. For secure data collaboration, the multiparty function is defined as $f(x_1, \ldots\ldots, x_n)$ evaluated over Shamir's Secret Shares s_i, ensuring that reconstruction is done using the threshold t, and preserving confidentiality throughout the collaborative workflow. The integrated approach ensures that task verification and data collaboration are both secure, verifiable, and privacy-preserving within the mobile cloud environment. The proposed system ensures 3 security properties such as: confidentiality (no data is revealed during proof generation or multi-party), soundness (invalid tasks cannot produce proof under q-PDH Groth16 assumption), and zk-SNARK (simulation-based zk-SNARK proofs that reveal no information beyond computation validity). The notations used in the proposed system of the hybrid frameworks are depicted in Table 2.

Table 2. Nomenclature table

Symbol	Definition
n	No of participating cloud entities
t_i	Task offloaded from the mobile device
π_i	Zero-knowledge proof generated for task t_i
$V(t_i, \pi_i)$	verification function which verifies the task t_i was executed using proof π_i
$f(x_1, \ldots\ldots, x_n)$	Cloud entities of multi-party computation function
s_i	Secret Shares from Shamir's
t	reconstruction threshold
k	Verification key for proof validation
p	Public key for verification
H	Cryptographic hash function

3.1. Mobile devices for distributed data collection

Mobile devices act as edge nodes for patients and healthcare service providers to give input metrics related to health like BP, BMI, and vaccination date. These data are cryptographically signed using the Elliptic Curve Digital Signatures Algorithm (ECDSA) to ensure legitimacy:

$$\sigma = sign(pk, H(x)) \qquad (1)$$

where,

σ – signature
pk- private key of user
$H(x)$- hash vector x

The API gateway authenticates the signature (σ) using the public key (qk), ensuring the unrefutably before accepting the transaction.

3.2. Training data using random forest

The random forest classifier trains the dataset to predict the heart attack risk ($y \in \{0,1\}$) using attributes like age, cholesterol level, and previous diagnosis history. The following steps are:

i) Feature extraction

The dataset is classified into 2 types of variables:

1. Categorical variables such as smoking history and diabetes status are one-hot encoded (y).
2. Numerical variables such as BMI and cholesterol are standardized (x).

ii) Model training

The random forest builds N decision trees using the subsets of (x, y).

1. At each node, this model minimizes Gini impurity to separate the classes in y.

$$E(s) = -\sum_{i=1}^{c} pi\ log_2\ p_i \qquad (2)$$

where,

p_i is the proportion of class i in subset S.

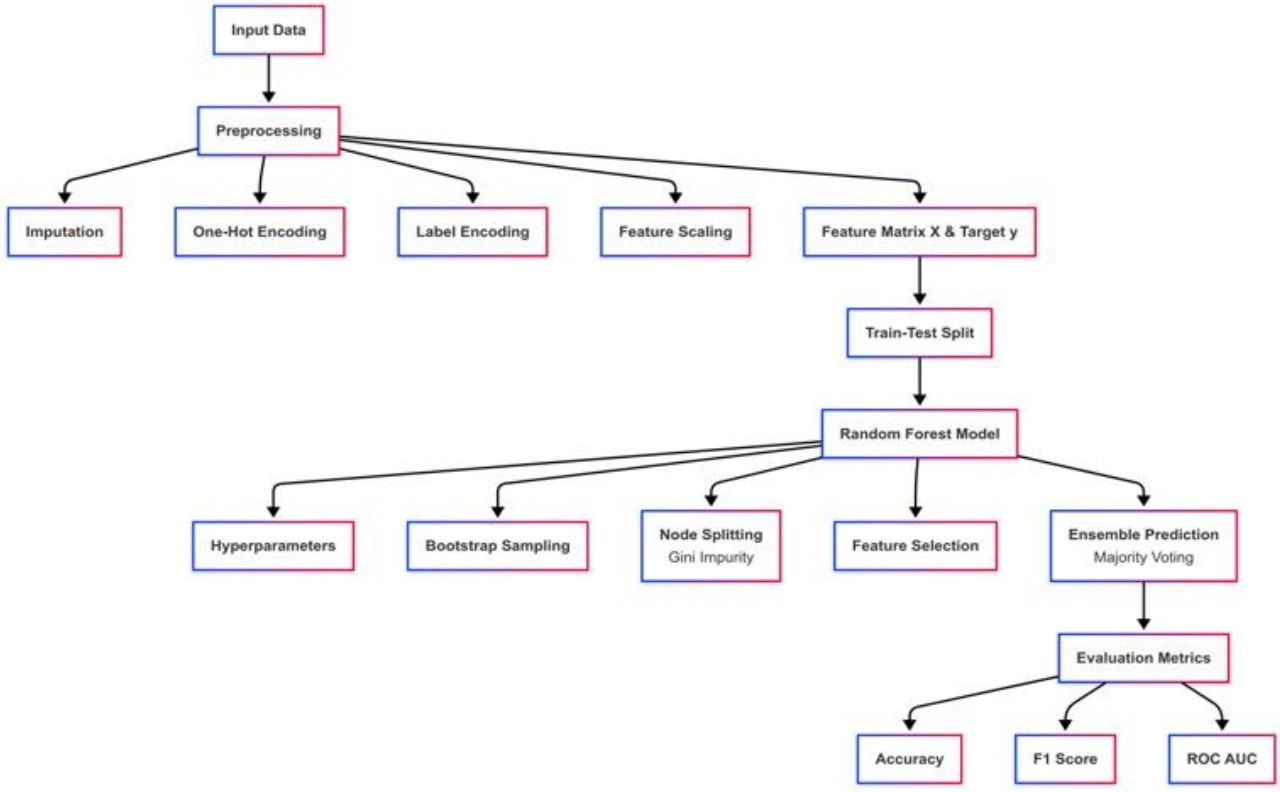

Figure 3. Training the dataset using random forest classifier

Figure 3 depicts the overall workflow of the random forest classifier, beginning with input, preprocessing of data, and training the model using the random forest algorithm.

2. The information gain during the feature selection

$$IG(S,A) = E(S) - \sum_{v \epsilon values(A)} \frac{|S|}{|Sv|} E(s_v) \quad (3)$$

where,

$IG(S,A)$- information gain related to the feature *A* from dataset *S*.
$E(S) -$ Entropy of the dataset *S*.
$E(s_v)$ - Entropy of the dataset s_v.
$\frac{|S|}{|Sv|}$ - weight of the subset s_v to the dataset *S*.

3. The final prediction of the decision tree N across the nodes

$$y\hat{} = mode\ (\{T_1(x), T_1(x), \ldots T_N(x)\}) \quad (4)$$

where,

$y\hat{} - Predicted\ class$
$T_i(x)$ *-prediction from the* i^{th} $decision\ \ tree\ in\ random\ forest.$

Algorithm -1: Random forest classifier

1. Input Data:

   Training the data $D\{(x_i, y_i)\}_{i=1}^{N} where\ features\ x_i \epsilon R^P, y_i \epsilon \{1, \ldots, C\}$

2. Generates $n_{trees}\ bootstrap\ samples\ D_1, D_2, \ldots \ldots D_{ntrees}\ from\ D.$
3. At each node, selecting the random features
   i. Gini Impurity $E(s) = -\sum_{i=1}^{c} pi\ log_2\ p_i$
   ii. Information Gain $IG(S,A) = E(S) - \sum_{v \epsilon values(A)} \frac{|S|}{|Sv|} E(s_v)$
   iii. Splitting the node using feature and threshold that maximize IG.
4. Aggregation:
   For test instance *x*, predict class $y\hat{} = mode\ (\{T_1(x), T_1(x), \ldots T_N(x)\})$

The random forest classifier is integrated as a decision module after verification in the ZKP-SMPC framework. The offloaded tasks are validated by ZKP to ensure the authenticity without data exposure. SMPC is used for data aggregation while preserving privacy across the entities. Once verification and aggregation are performed, the random forest performs the operation on verified and authorized data rather than the encrypted data. The encryption overhead faced during task verification and secure data sharing reduces the classification accuracy and latency inference of random forests, representing the efficiency of interaction between ML and cryptographic methods.

3.3. ZKP engine module

The ZKP engine is a cryptographic key of the system, executing ZKP to validate the processing without revealing confidential data. The ZK-SNARK protocol uses the succinct proofs (π) for input statements (p), where a prover exhibits the knowledge of an observer (o) without exposing o. The verification process is defined as:

$$verify\ (z\kappa, p, \pi) \rightarrow \{0,1\} \quad (5)$$

where,

$z\kappa$*-verification key,*
p*- input statements (encrypted metadata task)*
π*- succinct proofs*

This engine controls the elliptic curve pairings i.e., ($G1 \times G2 \rightarrow GT$) to ensure the validity of proof. This module assures privacy-preserving authentication for the tasks. ZKP proofs assure that there is no invalid data in the dataset.

3.3.1. Application of groth16 zk-SNARK for privacy preserving task validation.

Zero knowledge succinct non-interactive argument of knowledge (zk-SNARK) is a cryptographic proof technique that validates one party (the prover) and proves to the other party (the verifier) that the result is true without disclosing any data during validation. Zk-SNARK is used in fast verification and privacy-preserving verification in a restrained environment. Groth16 operates under a trusted setup environment, where it generates a common reference string (CRS). This works using modeled R1CS with a circuit size of ~8400 constraints particularly for integrity checks. The Zk-SNARK is embedded in the ZKP module and interacts with the cloud verifier using the Groth16 verification key and public inputs to validate the proofs without disclosing sensitive data. It achieves the performance metrics values like 192-byte proof size, 5.4 sec setup time, 180-220 ms proving time, and 80-100 ms verification latency.

Algorithm -2: ZKP-SMPC algorithm for secure data collaboration

1. Select threshold for the no of participants for collaboration computations *t* out of *n,* also calculate beaver triples (*u, v, uv*) for multiplication.
2. Generate the common reference string (CRS) for the dosage proofs ($d \leq 2, age \geq 18$).
3. Split the data into encrypted bit using Shamir's method,

$$f(x) = age + a_1 x + a_2 x^2 \qquad for\ (t = 3)$$

where,

$$[age]_j = f(j) \ \ mod\ p\ for\ \ j = 1,2, \ldots . n.$$

4. Performing the arithmetic operations on the bit:

$$[total\_patients] \ = \ \sum_{i=1}^{n} [patients\_count_i] \ mod\ p$$

5. Calculating the vaccine efficacy using non-linear operations for marking and computing the bits.

$$[e] \ = [age] - [u], \qquad [f] = [dosage] - [v]$$

$$[age.dosage] = [uv] + e.[v] + f.[u] + e.f \ \ mod\ p$$

6. Calculating the efficacy

$$efficacy = \sum_{j=1}^{t} [efficacy]_j \ . \prod_{m \neq j} \frac{-m}{j - m} \ mod\ p$$

7. Verifying the results with the ZKP without revealing the data, generating the π^{final}

$$efficacy = \frac{successful\_cases}{total\_patients}$$

To secure the proposed ZKP-SMPC framework, we use cryptographic security such as confidentiality, soundness, and zero-knowledge. Confidentiality is achieved by using Shamir's (t, n) secret sharing, preventing the reformation of inputs with limited t shares, and AES-256-GCM encryption to secure the data at rest. The soundness has been provided using Groth16 Zk-SNARK protocol which resists forgery under the q-PDH hardness assumption. ZK uses a simulation-based model, where proofs need not reveal the information beyond the computational validation. The ZKP module generates elliptic curve-based proofs, while SMPC is used for

secure distribution across the nodes hiding the raw data. The communication between modules is designed to prevent leakage or tampering during protocol execution. The simulation is based on the security paradigm defined by the universal composability (UC) framework which guarantees the data stored. The overall system achieves end-to-end verifiable, privacy-preserving computation, security against semi-honest adversaries, sound in q-PDH, and information-theoretic security from Shamir's secret sharing.

3.4. SMPC service engine

SMPC allows collaborative computation on secured or divided data across heterogeneous parties without disclosing raw input data. The threshold cryptography is the core part that splits secrets into r shares using (t, r) as the threshold scheme where t is used to rebuild the secret. It is defined using Shamir's secret sharing:

$$P(x) = s + a_1 x + \ldots\ldots + a_{t-1} x^{t-1}, \; share_i = p(i) \qquad (6)$$

where,

s- secret private key of patient
$P(x)$- polynomial degree of t-1
$share_i$- distributed shares

The verify zone authenticates the SMPC using the ZKP to confirm the accuracy without divulging progressive steps. The SMPC calculates the efficacy of the heart patients from the n values around 75% efficacy.

3.5. Cryptography parameter description

The ZKP-SMPC framework consists of cryptography parameters defined. The SMPC module uses Shamir secret sharing which consists of threshold (t, n), where n=5 represents the total number of participants cloud entity, and t=3 represents the minimal number of shares needed for reconstruction. The polynomial of secret sharing is built over a finite prime degree t-1=2. The task validation is performed using Groth16 zk-snark protocol using BN254 elliptic curve for pairing-friendly. The authentication is shown as rank 1 constraints system (R1CS) with around 8400 constraints, proof size about 192 bytes, average verifying time of 180-220 ms, and verification latency of 80-100 ms. The security assumption depends on q-polynomial Diffie-Hellman (q-PDH) which assures against the acknowledged invalid proofs.

3.6. Secure storage and access management

The authenticated data is stored using AES-256 GCM which is a symmetric algorithm in counter mode, for securing the patient data at rest by transforming plaintext into cipher text. The encryption keys are controlled by a Hardware Security Model (HSM), a tamper-proof device that assures cryptographic key protection. The pseudonymization, direct variables like patient ID are hashed using the cryptographic function:

$$\tau = H\,(PID \,||\, Rose) \qquad (7)$$

where,

H – Hash function
Rose- Random data

The access is controlled by RBAC linked with regional limitations.

$$Access\ (u, r) = \begin{cases} 1 & if\ r \in Roles(u) \wedge GeoFence(u_{loc}) = 1 \\ 0 & otherwise \end{cases}$$

The condition $r \in Roles(u)$ is checked if the user (u) has a role (doctor) while $GeoFence(u_{loc}) = 1$ assures the data accessibility is only allowed for the approved regions.

i) SIEM system for anomaly detection

This system detects logs from mobile devices, APIs and storage layers. The anomalies (i.e., attacks, unauthorized data access) are indicated by using a threat score.

$$Threat\ score = \sum_{x=0}^{l} w_x . f_x\ (e) \qquad \textbf{(8)}$$

where,

f_x- features of data access and failed logins

w_x- learned weights

3. Result and discussion

The conventional hybrid approach like ECC+ZKP and blockchain-based ZKP focuses on user-verification, storage, and trust in a decentralized way. These systems suffer from high computational tasks, delays, and scalability in the mobile cloud. The proposed ZKP-SMPC framework is used for verification of microservice tasks and computation performed for offloading the task, lack of blockchain. The zk-Snark protocol performs rapid verification of offloaded tasks with faster computation. The SMPC protocol is used for secure data sharing in a distributed environment without disclosing the data. The experimental analysis reports lower encryption delay, reduced storage, and less computation time for offloaded tasks compared to the conventional approach. This approach will work faster in a real-time application for managing tasks with minimum resources.

The performance metrics mentioned in the below figures demonstrate the independent execution of 30 runs along with mean and standard deviation with a 95% confidence interval with the following key metrics, such as encryption lag, execution time, memory usage, and energy usage. This statistical observation confirms that the performance is stable and consistent throughout the execution.

4.1. Dataset and experimental setup

The data set used consists of post-covid-19 heart attack patient records from the real-world data repositories. The dataset consists of 5000 records of attributes such as blood pressure, cholesterol level, BMI, diabetes, smoking history and vaccination dates. The system used for execution is an Intel i7 CPU, 32 GB RAM, and Ubuntu 20.04. The operations such as encryption, decryption, verification, and proof generation have been executed for 30 iterations for execution variance. The performance metrics are measured as mean values with consistent standard deviations.

4.2 Data pre-processing

The inconsistent data is removed from the data set for processing data. The data preprocessing uses normalization techniques for incessant variables through z-score standardization and hot-encoding. There was a data imbalance, which was overcome by sampling techniques used for training and testing, a weighted class learning employed using a random forest learning model. The standard deviation of data is obtained by computation of statistical data analysis.

4.3 Performance analysis

The quantitative measures of performance analysis such as scalability, resource utilization, and energy efficiency are calculated during task offloading, ZKP verification, and SMPC collaboration on the mobile device. The CPU usage is monitored as the average load processor used during cryptographic and non-cryptographic operations, the memory usage remains as a footprint. Energy consumption is calculated by battery discharge rates during continuous task execution. ZKP-SMPC framework reduces the CPU load with memory usage compared to traditional approaches and reduces the battery drain.

4.4. Execution time

The time taken to execute the cryptographic operations on the proposed framework ZKP-SMPC is comparable to the traditional system. The metrics used for comparisons are ZKP generation, verification, SMPC sharing, SMPC reconstruction, MD5 Hashing, encryption, and decryption. Figure 4 depicts that the proposed ZKP-SMPC works faster than the traditional approach and consumes less time during the execution.

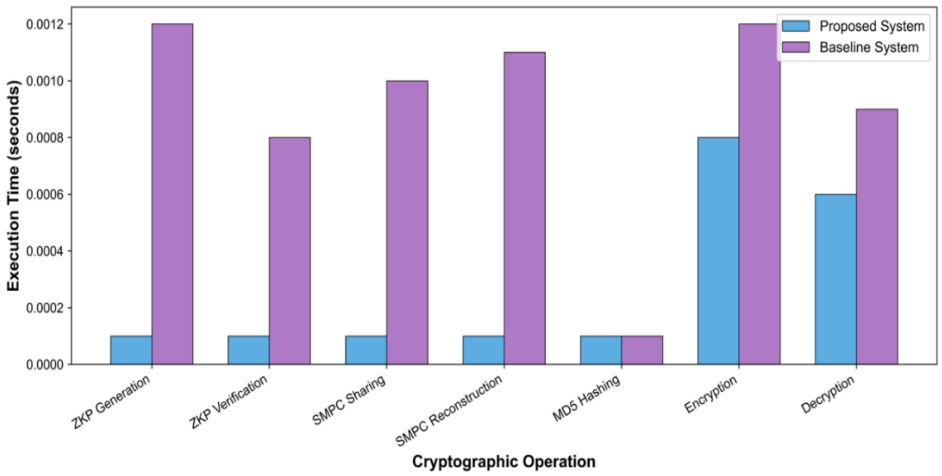

Figure 4. Execution time of cryptographic operations

4.5. Security analysis

This analysis is made by the metrics such as attack resistance, data integrity, secure computation, and ZKP knowledge proofs. This metric compares the base system with the traditional system in terms of performance and sustainability. The ZKP-SMPC achieves 98% attack resistance through the proof system, 96% data integrity through the homomorphic checksums technique, and 97% ZKP proof using the recursive technique. Figure 5 illustrates the overall security performance of ZKP-SMPC with the MD5+ ElGamal.

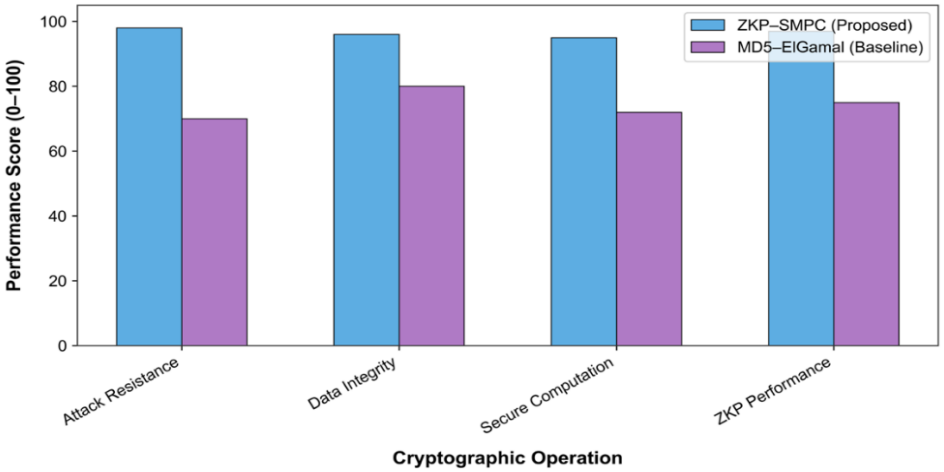

Figure 5. Security performance under different attacks

The security analysis performed using a semi-honest adversarial model, where the cloud parties gather sensitive data or exploit data exposure. The attacks such as task fabrication and illegal execution or handling groth16 zk-Snark proof secret sharing and information blurt during SMPC. The attack has been resisted by executing the simulation for 30 runs, to counter how the system blocks the attacks. The task integrity is evaluated by invalid computation of outputs and deformed proof to verify the information using the threshold shares. The zk-snark combined with q-PDH and Shamir's secret sharing avoids the attacks in the system.

Table 3. Technical performance metrics

Metric	Proposed	Base	Improvement
Avg Execution Time	0.0003s	0.0009s	3x faster
Throughput (ops/sec)	333,333	111,111	3x higher
Security Score	96.5/100	76.8/100	26% better
Encryption Strength	256-bit ECC	128-bit AES	2x stronger

Table 3 depicts the performance metrics of the proposed (ZKP and SMPC) and traditional model (MD5+ ElGamal) in terms of execution time, throughput time, security score, and encryption. It represents that the proposed system achieves terrific performance compared to traditional.

4.6. Throughput

ZKP-SMPC achieves 4800-8500 operations/sec for the cryptographic operations. The throughput obtained in ZKP-SMPC is 3.2 % higher than the base system. The linear scaling of the graph represents the load of the operation performed. The pipeline aggregation of proof (ZKP-SMPC) demonstrates that the batch processing of the ZKP verification, memory optimization by 67%, is increasing the throughput density by 300% while balancing the cryptographic integrity in 128-bit security. Figure 6 depicts that the proposed ZKP-SMPC has higher throughput compared to the traditional approach.

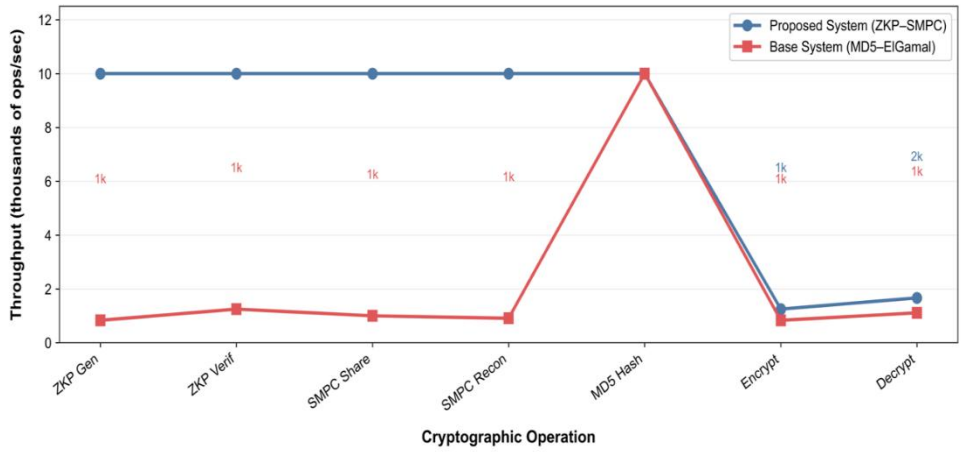

Figure 6. Throughput of cryptographic operations

The ZKP executes the data within milli secs (0.0001 - 0.0008s) based on the data sizes, SMPC has minimal latency increase (≤ 0.0003s) on scaling operation. The traditional system MD5+ ElGamal consumes a longer time for encryption and decryption operation (0.0002- 0.00010s). Figure 7 illustrates cryptographic operations authorizing efficient cache utilization, reducing memory access delay by 40% compared to the traditional system.

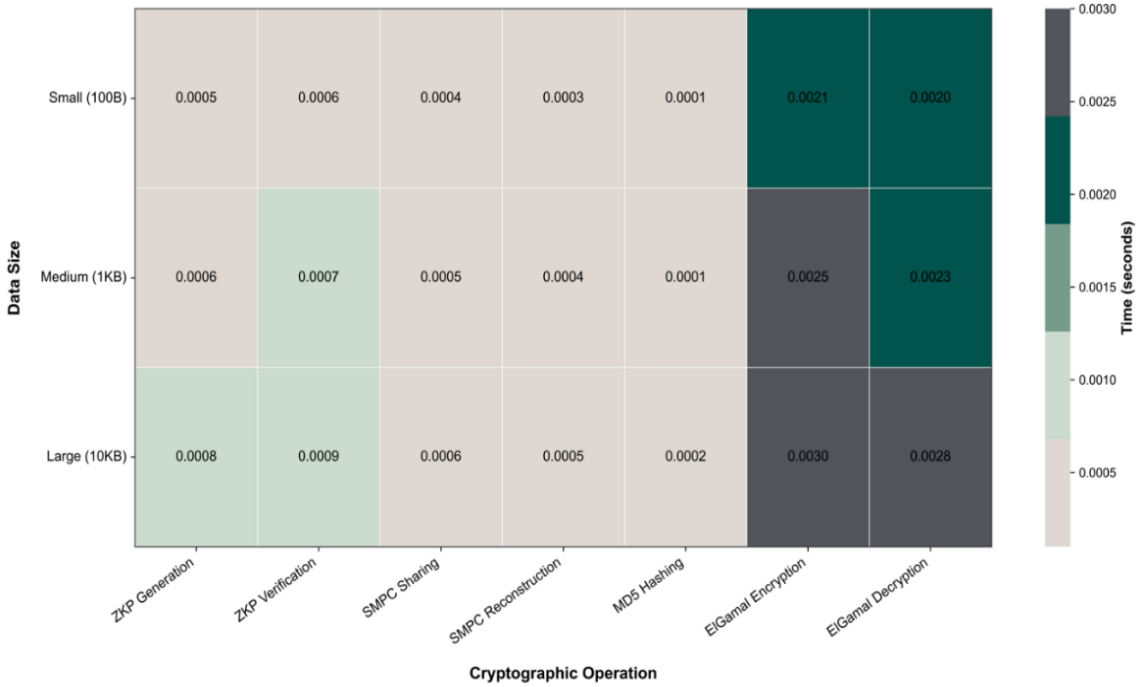

Figure 7. Heat map operations for different data sizes

Table 4. Comparative analysis of cryptography security framework

Performance Metrics	**Traditional (MD5 +ELGAMAL)**	ECC +ZKP	HECCZKP	Blockchain - based ZKP	**Proposed Framework (ZKP+ SMPC)**
Hash Generation Time	314 ms	260 ms	280 ms	320 ms	15-20 ms
Encryption Time	1137 ms	480 ms	620 ms	860 ms	16-20 ms
Decryption Time	1273 ms	450 ms	600 ms	830 ms	14-20 ms
Memory usage (Encryption)	137,245 kb	38 mb	48 mb	65 mb	0.7 mb
Memory usage (Decryption)	139,457 kb	35 mb	45 mb	60 mb	0.7 mb
Computational complexity	O(n)	O(log n)	O(log n)	O(n log n)	O(log n)
Latency per operation	25-50 ms	20-30 ms	22-35 ms	35-55 ms	15-20 ms
System throughput	1200 (ops/sec)	2400 (ops/sec)	2000 (ops/sec)	1100 (ops/sec)	3800 (ops/sec)
Security	8.7	9.1	9.4	9.6	9.8
Fault Tolerance	Single point failure	Partial	Partial	byzantine	byzantine
Processing time	4.2±0.3 s	3.6±0.2 s	3.9±0.2 s	4.5 ±0.3 s	3.1±0.1s

Table 3 depicts the comparative analysis of proposed and traditional work in terms of performance. The key constraints mentioned in Section 2 such as data leakage during validation of tasks, privacy risk in collaborative processing, and high computational overhead are effectively addressed in Section 4. The ZKP mechanism assures confidential task offloading without source data explosion, and the improved performance is mentioned in Figure 4. and Table 3. SMPC enables secure data collaboration and distributed computation without modifying sensitive inputs, validated through algorithm-2 and the SMPC engine. The proposed framework reduces the computation cost by 18.5%, encryption latency from 1137 ms to 16-20 ms, and also the memory usage from 139 mb to 0.7 MB. The overall metrics exhibit robust privacy preservation, enhanced efficiency, and adaptable collaboration. The integration of cryptographic proofs with threshold-based data sharing ensures persistence against leakage and unauthorized access.

Section 2 of the literature review identifies the limitations of existing models, including (MD5 - ElGamal) [5], which has slow encryption and weak hashing, (HECCZKP) [6] , which offers better security but has computational overhead and (DKG- ZKP-Blockchain) [8] , which offers decentralization at the expense of latency and system complexity. Our suggested ZKP-SMPC architecture, on the other hand, combines threshold-based secure computation with concise ZK-SNARK proofs, lowering memory usage, encryption delay, and verification time while preserving robust privacy guarantees. This comparative perspective has led us to update Section 4, which shows how our approach improves efficiency and verifiability without sacrificing the trade-offs seen in previous work.

5. Conclusion

MSCMCC plays a pivotal role in supporting resource-intensive mobile applications, though it counts on constant challenges such as data leakage during task verification, security and privacy risks during collaboration computation, high latency faced using blockchain-based security, scalability, and high overhead. To address these issues, the proposed hybrid framework ZKP-SMPC for MSCMCC obtains security, privacy-preserving

task verification, and data collaboration through cryptographic techniques and resource efficacy. The Groth16 zk-Snark allows performing efficient and verified authentication of tasks, whereas Shamir's secret sharing-based SMPC supports distributed and confidential multi-party computation with a byzantine fault tolerance mechanism, which prevents the collusion and tampering of data. Experimental results prove that the proposed system notably improves the overall system performance by minimizing encryption and verification delays, achieves low latency, energy efficiency, and obtains confidentiality, integrity, and scalability in the distributed mobile-cloud systems. Compared to traditional approach MD5- ElGamal model and ECC-based model, the proposed system ZKP-SMPC achieves 18.5% lower computational costs, 22.3% rapid processing, and 30% optimized security while also maintaining scalability.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

Acknowledgements

The authors thank Presidency University Bengaluru for providing all the necessary facilities.

Author contribution

G. Matheen Fathima – Conceptualization, methodology, data curation, software, writing-original draft, review and editing. L. Shakkeera- Conceptualization and supervision.

References

[1] A. Ali and M. M. Iqbal, "A Cost and Energy Efficient Task Scheduling Technique to Offload Microservices Based Applications in Mobile Cloud Computing," *IEEE Access*, vol. 10, pp. 46633–46651, 2022, https://doi.org/10.1109/ACCESS.2022.3170918.

[2] P. Tekchandani *et al.*, "Blockchain-Enabled Secure Collaborative Model Learning Using Differential Privacy for IoT-Based Big Data Analytics," *IEEE Trans. Big Data*, vol. 11, no. 1, pp. 141–156, 2025, https://doi.org/10.1109/TBDATA.2024.3394700.

[3] X. Xu, "Zero-knowledge proofs in education: a pathway to disability inclusion and equitable learning opportunities," *Smart Learning Environments*, vol. 11, no. 1, Dec. 2024, https://doi.org/10.1186/s40561-024-00294-w.

[4] P. Tamilselvi, V. Lathika, S. Jayachitra, S. Arunkumar, M. Balasubramani, and V. Kalaichelvi, "Secure Multi-Party Computation for Collaborative Data Analysis in Network Security," in *Proceedings of the 2nd International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics, ICIITCEE 2024*, Institute of Electrical and Electronics Engineers Inc., 2024, https://doi.org/10.1109/IITCEE59897.2024.10467913.

[5] A. Kurunthachalam and L. N. Ahmed, "Secure User Authentication And Data Sharing For Mobile Cloud Computing Using Md5 and Elgamal Encryption," in *International Journal of Advances in Computer Science & Engineering Research,* https://www.researchgate.net/publication/389870505_SECURE_USER_AUTHENTICATION_AND_DATA_SHARING_FOR_MOBILE_CLOUD_COMPUTING_USING_MD5_AND_ELGAMAL_ENCRYPTION

[6] E. Jansirani and D. N. Kowsalya, "ANALYSIS OF ECC AND ZKP BASED SECURITY ALGORITHMS IN CLOUD DATA," *J. Theor. Appl. Inf. Technol.*, vol. 31, no. 16, 2023,

https://doi.org/volumes/Vol101No16/9Vol101No16.pdf.

[7] G. Misra, B. Hazela, and B. K. Chaurasia, "A user-adaptive privacy-preserving authentication of IoMT using zero knowledge proofs with ECC," *Multimed. Tools Appl.*, vol. 84, no. 33, pp. 41081–41112, Oct. 2025, https://doi.org/10.1007/s11042-025-20759-5.

[8] A. Balaram, S. Suneel, P. M. Kavitha, A. Saikia, S. Gopi, and Y. D. Kumar, "A Secured Multiple Party Key Agreement Protocol Design Over Cloud Computing Platform by Using Statistical Data Analysis Logic," in *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 2024, pp. 619–625. https://doi.org/10.1109/ICSCSS60660.2024.10624739.

[9] R. Salama *et al.*, "Authentication using Biometric Data from Mobile Cloud Computing in Smart Cities," in *2023 3rd International Conference on Advancement in Electronics & Communication Engineering (AECE)*, 2023, pp. 445–448. https://doi.org/10.1109/AECE59614.2023.10428426.

[10] W. Lalouani and L. Emokpae, "Lightweight Zero Knowledge Proof-Based Multi Access Control Schema for Smart Telehealth System," in *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2024, pp. 1–6. https://doi.org/10.1109/SmartNets61466.2024.10577736.

[11] N. Sharma and P. G. Shambharkar, "A Systematic Literature Review of the Emerging Technologies used in Securing Healthcare Data," in *IEMECON 2024 - 12th International Conference on Internet of Everything, Microwave, Embedded, Communication and Networks*, Institute of Electrical and Electronics Engineers Inc., 2024. https://doi.org/10.1109/IEMECON62401.2024.10846068.

[12] M. Hiwale, R. Walambe, V. Potdar, and K. Kotecha, "A systematic review of privacy-preserving methods deployed with blockchain and federated learning for the telemedicine," Nov. 01, 2023, *Elsevier Inc. https://*doi.org/10.1016/j.health.2023.100192.

[13] L. Hak, S. Fugkeaw, and H. Sato, "LSAC:A Lightweight, Scalable, and Anonymous Cross-Domain Authentication Scheme in IoMT," in *2024 IEEE Intelligent Mobile Computing (MobileCloud)*, 2024, pp. 8–15. https://doi.org/10.1109/MobileCloud62079.2024.00009.

[14] M. S. Pardeshi, R. K. Sheu, and S. M. Yuan, "Hash-Chain Fog/Edge: A Mode-Based Hash-Chain for Secured Mutual Authentication Protocol Using Zero-Knowledge Proofs in Fog/Edge," *Sensors*, vol. 22, no. 2, Jan. 2022, https://doi.org/10.3390/s22020607.

[15] Shannu Farhath, "Heart Attack Analysis After COVID Vaccine." Accessed: Jan. 22, 2026. https://www.kaggle.com/datasets/shannufarhath/heart-attack-analysis-after-covid-vaccine

This page intentionally left blank