A Robust Lattice-Based Protocol for Outsourcing Scientific Computations in Multi-Cloud Architecture

Rajendra Asuri*[1], Santhosh Reddy BasiReddy [2],

[1] Covestro, Pittsburgh, USA
[2] USAA, USA

Emails: asurirajendra.sap@gmail.com, santhureddy245@gmail.com,

Abstract

The research delivers an enhanced lattice-based encryption protocol which operates specifically for scientific computing processing in multi-cloud systems. The system employs two independent cloud servers to provide secure data processing with privacy features and efficient operation. The users employ a probabilistic lattice-based encryption method to secure their private data while the scheme protects against quantum attacks and provides semantic security. The encryption method uses homomorphic transformations to process input data as encrypted messages that protect the original text alongside the decryption keys. The architecture supports parallel processing and dynamic scaling, enabling secure multi-user collaboration across heterogeneous computational environments. Performance evaluation of the protocol shows how it functions under different data size conditions from kilobytes to gigabytes and cryptographic key length settings from 16 to 256 bits. The proposed solution outperforms customary homomorphic systems by reaching a 35-40% reduction in operational cost with high-throughput scenarios resulting in a 50% shorter upload time and decryption duration. Both multilevel encryption and the decryption procedure operate in phases which permits authorized users to recover final data while blocking unauthorized access for confidentiality and proper access control. The model enables substantial enhancement of reliability and fault tolerance and security for real-time applications running across multiple cloud infrastructures.

Keywords: *Lattice-Based Encryption, Multi-Cloud Architecture, Scientific Data Processing, Homomorphic Encryption, and Secure Computation Outsourcing.*

1. INTRODUCTION

Cloud computing stands as an essential component in modern data processing because it enables organizations to obtain effective computational resources and adjustable storage facilities [1]. Organizations together with researchers maintain their dependence on cloud services for complex computation outsourcing yet they now face three vital challenges regarding data privacy alongside computational integrity standards and multi-cloud connection problems [2]. Working with semi-trusted cloud servers to execute tasks without disclosing important information has become the foundation for distributed computing security.

Cloud computing lets both single researchers and worldwide business organizations allocate their demanding computations to external third-party computing centers. The practice of outsourcing delivers both economic benefits and scalability improvements to users. Computations that surpass local management introduce various security risks which include unauthorized access of data and integrity breaches and malicious alterations of computational outcomes [3]. The security risk escalates when organizations work with multiple cloud providers in multi-cloud architectures because it expands their total vulnerable infrastructure area and makes it harder to handle their security needs.

Data Security and Outsourcing Challenges

The main problem in computation outsourcing centers on establishing secure data privacy combined with unalterable result authenticity and data integrity maintenance. Cloud service providers should not be considered reliable because customers need to validate that data entry remains secret while ensuring result accuracy from providers. Data security problems become more severe because medical

records and proprietary research accompany financial transactions [4]. The Figure 1 represents the standard process of outsourcing by showing users giving encrypted data to cloud servers for processing and receiving encrypted results. Verifying the accuracy of such results without local recreation of the computations proves to be non-simple. Users need methods to check that their computation tasks were properly executed.

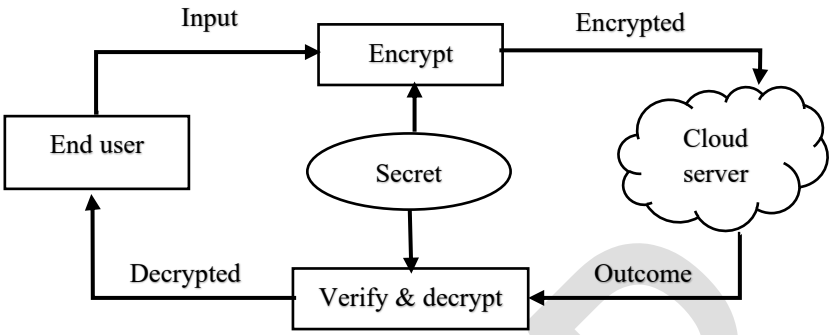

Figure 1: Secure Outsourcing Scenario

The Figure 1 illustrates the secure outsourcing model through which users transmit encrypted data to cloud servers which process the data with encryption then send back the results for user decryption. Ensuring correctness together with secrecy and verifiability remains the main challenge across the entire process.

Role of Homomorphic Encryption

Secure outsourcing problems find a potential solution through Homomorphic Encryption which provides secure computation on encrypted data. Such encrypted computational operations will only reveal their results to the data owner because he possesses the decryption keys [5]. The protected nature of computations performed by the cloud server depends upon its exclusive ability to deal with encrypted information rather than the underlying data. FHE stands as an especially strong form of HE among all its different versions. FHE allows processing of encrypted data through both additive and multiplicative operations on ciphertexts. FHE developed the core principles required to establish secure cloud computing platforms where tasks delegate their executions through protected systems.

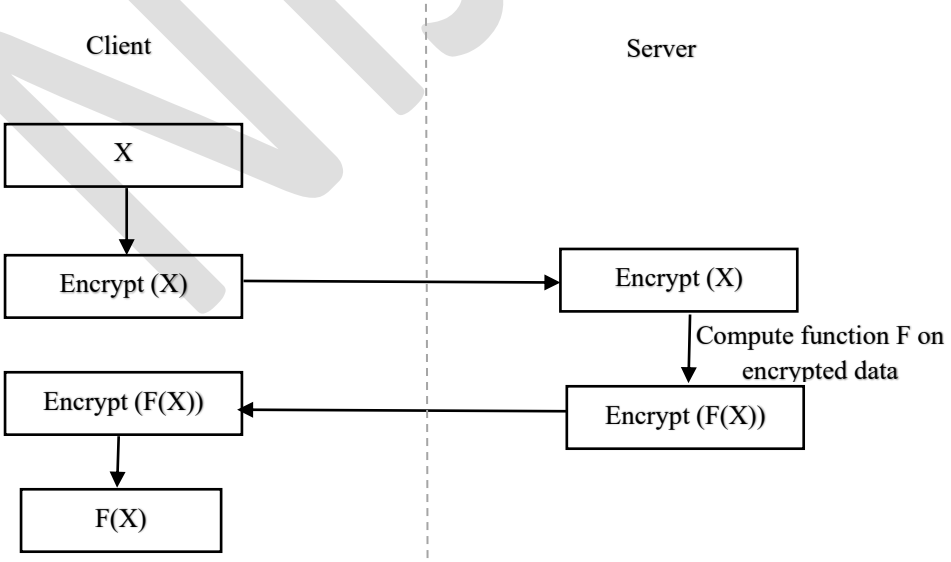

Figure 2: Illustration of Fully Homomorphic Encryption System

The security computations using FHE are shown in Figure 2. Security operations on encrypted values proceed in the cloud due to FHE without revealing information about original plaintext content to the service [6]. The user retrieves the outcome which effectively corresponds to calculations performed on the original values through decryption. The practical implementation of FHE faces challenges

because of the significant computational expenses associated with it. Scientific inquiry conducts research about dual-methods that unite basic cryptography with lattice-based encryption systems due to their dual advantages of quantum-resistant properties and performance improvements [7].

Lattice-Based Cryptography and Learning with Errors

Lattice-based cryptography based on SVP and the LWE problems has proven successful to build security protocols due to its mathematical challenge resistance. Lattice problems demonstrate strong resistance to quantum attacks because of which they serve as appropriate elements for modern post-quantum cryptographic systems [8]. Various cryptosystems use the LWE problem which remains challenging to solve even when quantum computers become operational. A challenge exists in solving linear equations with small random errors because this task cannot be accomplished within polynomial time. Investigations into such cryptographic systems allow users to share confidential data and establish secure computational delegation methods while building verifiable computation protocols. Lattice-based systems allow developers to build small yet powerful encryption methods that fulfill practical security requirements for outsourcing operations [9].

Multi-Cloud Landscape

The increasing use of multi-cloud environments that include services from AWS and Azure and Google Cloud creates new issues in connection security management combined with policy integration and security protection needs. Multiple heterogeneous infrastructures used for computation distribution bring separate operational models and threat vectors. Secure data encryption should be coupled with extensive authentication measures and the ability to control trust relationships between cloud providers as well as provide flexible infrastructure expansion capabilities [10]. Building security protocols requires achieving the ideal equilibrium between operations speed and encryption power and security user-friendliness. The current situation presents strong motivation for combining lattice-based encryption with homomorphic functionalities. This method enables the development of protected scalable computation procedures which verify their operations across multiple cloud environments [11].

Secure and efficient outsourcing of computation continues to grow because various fields use it for their work.

- Systems that use scientific computing methods solve large matrix operations together with systems of linear equations
- Secure data mining and machine learning
- Biometric and image processing
- Multi-party computations
- Financial analytics

All these application domains require secure offloading of complex tasks that preserve both data security and its protected state. Large linear algebra computations done by research institutes and encrypted patient record processing for healthcare providers through cloud-based systems operate with common security concepts [12]. This research creates a secure multi-cloud specific lattice-based protocol that resolves the multiple operational challenges of cloud computation outsourcing for scientific applications. The approach aims to bridge the gap between academic cryptography study and practical implementation requirements, particularly by emphasizing post-quantum security characteristics, accountability, and adaptability.

2. RELATED WORK

Cloud computing has brought about a major transformation of how computational operations are executed today. Users who have insufficient computational capabilities depend on cloud systems to execute their large and mathematically intricate scientific computations [13]. Critical issues emerge regarding database privacy together with system security as well as computation outcome accuracy when such operations are delegated to external services. The protection of data during scientific

outsourcing requires multiple solutions of which attribute-based encryption (ABE) and secure scientific outsourcing schemes and homomorphic encryption remain the most notable [14].

Homomorphic Encryption for Secure Outsourcing

The main method through which privacy-preserving computation functions is homomorphic encryption. Cloud servers can process encrypted data independently through homomorphic encryption technology which prevents them from accessing plaintext information. All critical information stays shielded from unauthorized access because the encryption system secures the information even when dealing with untrusted servers [15]. The previous security systems processed encrypted user inputs utilizing public key encryption throughout public cloud services. The returned results came encrypted to users who needed to decrypt them. Such systems proved secure but they added excessive costs and delays to cloud systems. Non-interactive methods solved practicality problems by minimizing the user-cloud communication process. The introduction of multi-key fully homomorphic encryption techniques in some schemes provides enhanced flexibility because it enables secure processing of data encrypted under multiple keys.

Attribute-Based Encryption and Access Control

Secure data sharing along with fine-grained access control in cloud applications adopts ABE as its main implementation technology. This method encrypts data according to attributes after which the decryption process becomes possible only when user attributes line up with defined access rules. The mode of operation which ABE supports work specifically well in collaborative projects [16]. The development of ABE extensions targets two performance-related goals including revocation system support alongside verification procedures for decrypted data integrity. These methodologies force users to deal with computing challenges or prevent them from achieving complete protection of their data.

Outsourcing Scientific Computations

Multiple methodologies have been developed to securely outsource particular scientific operations which include matrix multiplication and sequence comparison and linear programming. Laboratory techniques implement cover-unlocking strategies together with protected information splitting alongside protocol systems that support both secure result computations and user data security. Numerous tactics to address security flaws exist in computing but still contain multiple operational restrictions. The analysis of semi-honest cloud models requires an assumption that cloud servers abide by protocols but attempt to extract user private data [17]. The real-world operations of server systems produce counterproductive results since the server operates maliciously either by providing erroneous data or intentionally extending computation times. The execution of certain procedures demands consecutive interactions taking place between servers and clients during multiple phases. Large computations that depend on multiple server-round interactions result in performance reduction. Matrix-vector operations require partitioning of large excessive matrices exceeding available memory sizes because of which they generate additional operational overhead and complexity [18].

Table 1: Overview of cloud computing's safe outsourcing methods

Method	Approach	Advantages	Drawbacks
Fully Homomorphic Encryption (FHE)	Computation on encrypted data [19]	High privacy; secure outsourcing	High computational cost; impractical for large-scale tasks
Attribute-Based Encryption (ABE)	Policy-based access control	Flexible; revocation supported [20]	Can be inefficient for large datasets; overhead in key updates
Masking Techniques	Data obfuscation	Simple implementation	Risk of data leakage; low robustness [21]

Matrix Multiplication Outsourcing [22]	Scientific computing offloading	Efficient for small-scale problems	Memory-intensive; impractical for large matrices
Multi-key FHE	Bootstrapping over ciphertext	Supports complex operations [23]	Complex setup; increased latency
One-Time Pad Encryption	Bit-level encryption [24]	Strong secrecy if used correctly	Susceptible to known-plaintext attacks; pad reuse issue
Lattice-Based Scheme	Multi-cloud, non-interactive	High security; resistant to quantum attacks	Requires lattice complexity understanding [25]
GPU-Based Homomorphic Scheme	Parallel computation [26]	Speedup; scalable with core increase	Requires GPU infrastructure
Secure Linear Programming (LP)	LP solver outsourcing	Secure boundary separation	Guess-and-determine attacks possible [27]
Boolean Encryption Scheme	Basic bit-wise file encryption	Low computational cost [28]	Weak security for complex problems

Challenges and Drawbacks in Existing Systems

Most current models experience operational weaknesses although substantial advancements have occurred. Certain encryption protocols depend on semi-honest cloud models which demand the cloud services to perform protocol actions while attempting to gain knowledge about the shared data. Such an assumption fails to recognize the intentional destructive outcomes which a malicious server can perform through altered responses or delayed reaction times [29]. The execution of several rounds between clients and cloud servers exists as a fundamental requirement of numerous secure computation protocols. The communication requirements for handling large matrix operations with extensive computation quickly become impractical. The multiplication process for matrices and vectors consumes high memory resources because it forces inputs to be divided into separate processing stages through multiple passes. Several instances of one-time pad encryption systems have faced criticism for being ineffective. Such encryption protocols need a pad with dimensions matching their message while requiring pad utilization in just one instance and secure sharing among the parties. Known-plaintext attacks allow attackers to guess several ciphertext bits then reconstruct additional parts through pattern recognition of the keystream.

Old security systems for cloud computing evolved through traditional encryption until researchers developed homomorphic and lattice-based schemes [30]. Various improvements have occurred yet existing obstacles persist mainly because of the equilibrium needed between system performance and security features and scalability potential. A table 1 in presents the main advantages and shortcomings of different methods along with data that can guide research into secure scientific computation outsourcing.

3. PROBLEM STATEMENT

Cloud platforms now serve as essential solutions to process computationally extensive operations due to both scientific data growth and performance requirements in computing. Due to its deployment this model presents substantial security as well as privacy-related issues. Single-cloud deployment models expose scientific data to privacy risks because they face security issues like data breach incidents combined with the risk of unauthorized system access as well as externally-triggered single-points-of-failure breakdowns. The application of conventional cryptographic methods falls short when processing enormous multi-user computations because they lack suitable performance along with scalability features.

Current solutions encounter one or multiple problems which include:

- High computational overhead in secure data processing.
- Inadequate support for multi-user collaboration and data privacy.
- A centralized cloud infrastructure lacks sufficient tools to stop attacks between users and system breaches.

Scientific computation requires an immediately necessary secure efficient scalable cryptographic protocol which enables privacy protection across multiple non-colluding cloud servers. This protocol must ensure:

- Robust protection of user data through advanced encryption techniques (e.g., lattice-based cryptography),
- The system should include support for secure operations that allow encrypted data processing.
- The system enables authorized users to access results but their performance remains low and operational delay is reduced to minimum.

4. PROPOSED WORK

The approach establishes a secure lattice-based computational outsourcing protocol which enables many users to send sensitive scientific computations to non-colluding cloud servers named CS1 and CS2. The framework implements lattice-based cryptography for encryption along with homomorphic transformation for protected ciphertext computation and multi-layered decryption to authenticate users who can obtain results.

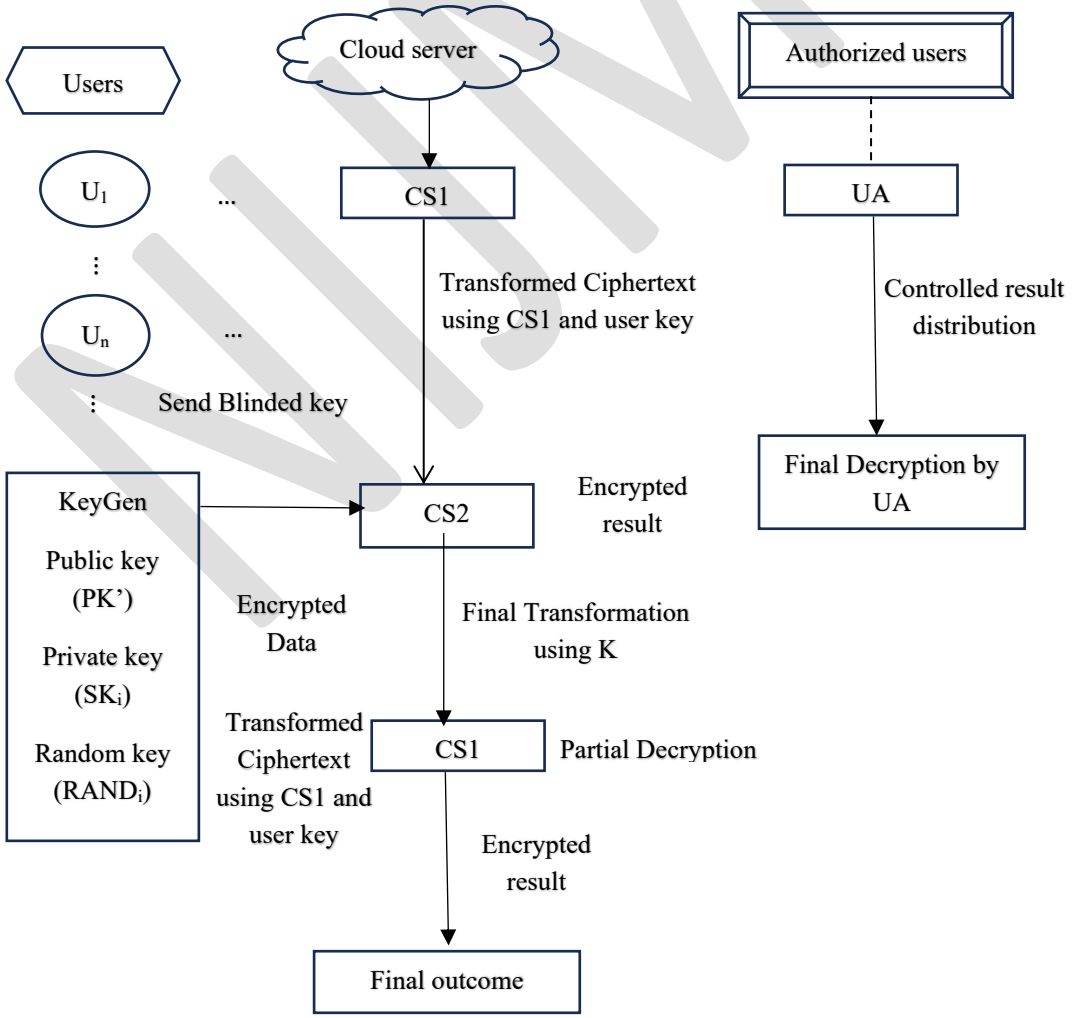

Figure 3: Flow illustration of suggested approach

The Figure 3 demonstrates the sequential process of the lattice-based protocol which secures scientific computation outsourcing when using multi-cloud architecture.

Key features:

- Multi-user and multi-cloud model.
- Post-quantum security via lattice-based encryption.
- Homomorphic circuit evaluation over encrypted data.
- Access control and non-collusion-based security.

Multi-User and Multi-Cloud Model

The proposed system operates in a multi-user model encompassing users from the set $\{U_1, U_2, \ldots, U_n\}$ who encrypt their private inputs and transmit them to non-colluding servers CS_1 and CS_2 which comprise a multi-cloud architecture. The model achieves better parallelism and distributed processing together with privacy enhancement.

The system depends on user inputs U_i which consist of $\alpha_i \in R_q$ that represent elements of $R_q = \mathbb{Z}_q[x]/(f(x))$. A user produces two cryptic elements with $SK_i \leftarrow \psi$ as the secret key after calculating a public key.

$$PK_i = a_i \cdot SK_i + 2\alpha_i \tag{1}$$

The procedure to protect α_i produces ciphertext pair $c_i = (c_{i0}, c_{i1})$.

$$c_{i0} = \langle u_i, e_i \rangle + a_i, \quad c_{i1} = \langle a_i, e_i \rangle + \epsilon \tag{2}$$

The ciphertexts are sent to CS1 as component ci and at the same time CS2 receives a blinded secret $\tau_i = RAND_i \cdot SK_i$. CS2 transforms it to:

$$\gamma_i = K_{CS2} \cdot RAND_i \cdot SK_i \tag{3}$$

which is relayed to CS1. Software servers operate independently of each other since they cannot access SK_i and c_i simultaneously thus maintaining data separation. Multiple users can actively participate within the model at the same time as it distributes computations securely by keeping user inputs and keys secret.

Post-Quantum Security via Lattice-Based Encryption

Lattice-based encryption functions as the fundamental element of this protocol since it has been developed to withstand quantum attack methods. The system depends on difficult security problems including the Learning with Errors (LWE) algorithm combined with Short Integer Solution (SIS) in complex lattice structures. Secure lattice-based encryption schemes are believed to stay hard even after the development of quantum computing technology.

Ring-LWE-based encryption scheme uses the polynomial rings $R_q = \mathbb{Z}_q[x]/(f(x))$ where both messages and keys remain for encryption. The ring features elements of Z_q and f(x) as a cyclotomic polynomial and q as a large prime modulus. The standard process for encrypting message $m \in R_q$ consists of:

$$Enc(m) = (u = a \cdot r + e_1, v = b \cdot r + e_2 + m) \tag{4}$$

Here $a \in R_q$ = public, $b = a \cdot s + e$ = second component of the public key, $s, r, e_1, e_2 \leftarrow \chi$ = sampled from a noise distribution χ, m = embedded as a small polynomial.

The security system enables semantic protection through its ability to encrypt both messages and secret key structures. The lattice structures contain homomorphic properties that enable secure operations on ciphertexts without decryption operations. The proposed method protects all sensitive dataset information including inputs and intermediate and final outcomes using encryption from a lattice-based scheme which makes it quantum-resistant throughout its computational processing.

Homomorphic Circuit Evaluation Over Encrypted Data

The proposed protocol demonstrates an essential capability which enables the assessment of functions on encrypted data through homomorphic operations without the requirement to decrypt this information. The cloud must execute $f(\alpha_1, \ldots, \alpha_n)$ without acquiring knowledge about the α_i terms to support secure outsourcing operations.

The Boolean/arithmetic circuit ξ_{FUN} contains only addition (+) and multiplication (×) gates which perform the computations. With transformed ciphertexts $c_i^{(2)} = (K \cdot c_{i0}, K \cdot SK_i \cdot c_{i1})$, CS2 performs gate evaluation through its process.

Addition Gate:

For users i and j,

$$c_{add} = K \cdot \left[(SK_i \cdot c_{i1} - c_{i0}) + \left(SK_j \cdot c_{j1} - c_{j0}\right)\right] \quad (5)$$

Multiplication Gate:

$$c_{mul} = K^2 \cdot \left[(SK_i \cdot c_{i1} - c_{i0}) \cdot \left(SK_j \cdot c_{j1} - c_{j0}\right)\right] \quad (6)$$

The evaluated output after all gates is:

$$\beta' = K^{\theta+1} \cdot \beta \quad (7)$$

The number of multiplicative layers is represented by θ within the formula. The result β remains hidden through the multiplication of transformation key values. The encryption method both preserves data accuracy through homomorphic transformation and protects information confidentiality because every operation works on secured data.

Access Control and Non-Collusion-Based Security

The system enforces access control by using a two-stage decryption process in which only authorized personnel can proceed. Under the premise of honest server operation CS1 and CS2 cannot achieve secret reconstruction or result decryption because they maintain independent functionality.

The computation concludes as CS2 obtains $\beta' = K^{\theta+1} \cdot \beta$ to forward it to CS1. The authorized user U_A needs CS1 to complete decryption procedures after implementation of the blinding process.

$$\beta'_A = RAND_A \cdot K_{CS2}^{\theta+1} \cdot \beta \quad (8)$$

CS2 removes its key:

$$\beta_A = RAND_A \cdot \beta \quad (9)$$

Finally, the user removes the blinding factor:

$$\beta = \frac{\beta_A}{RAND_A} \quad (10)$$

The encryption protocol allows only $RAND_A$ holders to derive β because of this protocol function. The unshared nature of $RAND_A$ between the cloud ensures that CS1 and CS2 cannot put together β even when they possess shared data since there is no complete reconstruction possible. Access control becomes guaranteed due to the non-collusion assumption which ensures security when used in sensitive defense and health applications together with financial modelling needs.

Table 2: Key Components & Notations

Symbol	Meaning
U_i	User i, with private input α_i
SK_i, PK_i	Private and public keys of user i
$RAND_i$	Random number for key obfuscation
CS_1, CS_2	Cloud Server 1 and Cloud Server 2
c_i	Encrypted ciphertext of user input α_i

Symbol	Meaning
K_{CS1}, K_{CS2}	Server-side secret values
β	Final computation result
ξ_{FUN}	Function circuit applied on inputs
θ	Number of multiplicative gates in ξ_{FUN}

Algorithm 1: Lattice-Based Secure Outsourcing in Multi-Cloud Architecture

Input:

- Private user inputs $\alpha_1, \ldots, \alpha_n$
- The parameters of the lattice include the modulus q and the ring $R_q = \mathbb{Z}_q[x]/(f(x))$ in addition to the error distribution ψ.
- The function circuit ξ_{FUN} needs to be assessed.
- Two non-colluding cloud servers: CS1 and CS2

Output:

- Result β securely accessible only by authorized users $UA \subseteq \{U_1, U_2, \ldots, U_n\}$

Step 1: User Key Generation and Cloud Initialization

- Every user individually produces their secret key and then calculates their public key while remaining their own key.
- Each user picks a unique random number which will be employed for later secret key encryption.
- Two cloud servers referred to as CS1 and CS2 independently generate secret transformation keys during their operations.

Step 2: Secure Sharing of Blinded Keys

- The blind process occurs through a multiplication of random numbers to specific secret keys by each user.
- The blinded key travels from the user to CS2.
- he blinded key reaches CS1 after CS2 processes it with an internal secret key.
- CS1 retrieves an altered version of the secret key through the random number selected by the user while maintaining privacy about the actual key.

Step 3: Encryption of Private Input

- The system requires each user to encrypt their private information through their public key accessibility.
- The processed data is transmitted to CS1 after encryption.
- The encryption protects the input from being accessible to the cloud system so it cannot see the original details.

Step 4: Ciphertext Transformation for Secure Computation

- CS1 applies an internal key to encrypt the data based on the transformed secret key it previously received.
- The transformed ciphertext proceeds to CS2 from CS1.
- CS2 adds a further encryption layer that enables the data to become ready for computational operations.

Step 5: Secure Evaluation of the Computation

- CS2 applies the functions (matrices or machine learning inference models) to the encrypted data which underwent full transformation.
- The cloud system operates all mathematical functions on data without performing data decryption procedures.
- The evaluation calculation result stays encrypted under multiple layers of protective transformation.

Step 6: Controlled Distribution of the Encrypted Result

- The system transfers the encrypted data to CS1.
- The result preparation process at CS1 generates user-specific data outputs through the application of random numbers assigned for each authorized access point.
- The information reaches CS2 where encryption sections are stripped away before forwarding the result to the user.

Step 7: Final Decryption by Authorized Users

- Authorizing users obtain individual versions of the encrypted result.
- Users gain access to the final output by applying the personal random number provided to them which removes encryption from the result.
- Those users lacking the right random key cannot read or decode the final output.

Phase 1: Key Generation and Initialization

The key generation phase requires each user to create secure keys through lattice-based encryption because this method remains resistant to both classical and quantum cryptographic attacks. The encryption method uses two essential keys; public keys are distributed to users yet they maintain secret their private keys. Security rests on mathematical lattice structures that present multi-dimensional point arrays. The security model relies on the complexity of design elements implemented during its development. Each independent cloud server generates separate transformation keys. These keys belong to CS1 and CS2. The keys created here will serve to conduct safe operations on encrypted information later. The initialization process deletes all possibilities for any party including cloud servers to recover user private information or decryption keys. The spread-out architecture stands as the operational framework connecting multiple users in a protected manner. This phase generates critical cryptographic assets that support encryption procedures and secure transformation as well as result delivery operations for following phases. The cloud servers function individually during all phases without conferring with others meaning users maintain complete privacy through their computation period. During this phase the system ends up prepared for computation outsourcing security without exposing any raw data or secret keys to external systems.

Phase 2: Blinding and Secure Key Exchange

The second phase protects privacy by providing users with the ability to communicate vital key data to the cloud platforms while maintaining their actual secret keys intact. Users make their private keys unreadable through random blinding factors which allow safe transfer to Cloud Server 2. By using blinded versions of information CS2 cannot decipher the original key even though it receives the data. The secure transformation process at CS2 executes with its private secret before sending results to first cloud server CS1. The blinding factor known to CS1 enables its ability to extract the usable transformed private key version while remaining unaware of its original form. Complete key secrecy remains at CS1 because the system allows this cloud server to execute user data operations. Independence between servers functions as the basis of this operation because coordinated actions could undo the blinding process. According to the protocol it does not expect CS1 and CS2 to work together as they both lack knowledge about the keys. A protocol separates knowledge from control functions during this phase to guarantee that none of the cloud servers obtains complete client information. The protocol enables secure private data processing through its mechanism that allows cloud servers to work indirectly with user keys.

Phase 3: Encryption of Private Input

The encryption process starts by having each user protect their private input data with a security operation that uses their public key. The encryption technique transforms plaintext through a lattice-based operation to generate ciphertext which contains multiple elements. The transformation process makes the input material completely unintelligible to any unwanted observers such as cloud servers. The users forward their encrypted data inputs to CS1 through the system. Thanks to lattice encryption protocols an attacker cannot decrypt intercepted ciphertext because they require the private key which makes decryption impossible. The encryption operation contains elements of randomness and noise that obscure original data patterns making analysis by unauthorized viewers impossible. During this phase sensitive scientific data consisting of numerical models and datasets maintains its privacy holder status while being processed on the cloud computing platform. Using the protocol enables calculations to proceed directly on encrypted values through homomorphic operations before decryption happens. Such protocol design maintains privacy through encrypted computations thus giving users a critical functionality. The encryption implementation proves to be scalable together with its lightweight nature and works efficiently through multi-user systems that support various data types in distributed cloud applications.

Phase 4: Homomorphic Transformation of Ciphertexts

The transformation process at CS1 begins once it receives encrypted inputs from users to prepare data for secure computation. The system applies operations to each ciphertext using its private transformation key and acquired transformed user key to keep the data structure appropriate for future homomorphic evaluation. During this stage the actual content of the data stays unchanged in its encrypted form yet the data is adjusted for efficiency and secure computation purposes. The transformed ciphertexts travel from CS1 to CS2 where it undergoes additional transformation processing. Through these combined operational steps the required control structures are embedded into encrypted data which allows forthcoming homomorphic operations without exposing the actual data contents. Both private keys and user secrets are necessary for servers to reverse transformations because the servers do not have access to these authentication standards. The entire collaborative transformation procedure acts as the foundation for upholding functional capabilities and privacy. Computing operations on the encrypted data become possible through this method because it maintains hidden data values. The execution of encrypted data between clouds during this phase plays a fundamental role in advancing a completely privacy-protecting multi-cloud computing system.

Phase 5: Homomorphic Evaluation of the Function Circuit

The actual calculation takes place in this phase though it remains encrypted at all times. From this point CS2 performs the evaluation of an encrypted function by utilizing the transformed ciphertexts. Because of its lattice-based encryption capabilities it becomes possible for the system to directly execute both addition and multiplication operations on encrypted data. The cryptographic operations duplicate the calculations which regular users conduct for statistical work and machine learning operations along with numerical simulations. At every gate stage the output remains completely encrypted while each computation step retains full confidentiality of the input data. Multiple layers of computations can stack up as the protocol functions as a bounded circuit depth homomorphic system. When the final gate computation finishes its encryption, it gets enveloped by transformation keys which render the result unreadable to unauthorized users. Parallel processing is possible with the circuit model because it enables simultaneous evaluation of various ciphertexts and gates. The stage demands high computational power although it preserves total security throughout the process. The secure outsourcing method provides scientists with a way to protect their data and operations from revealing any information—including intermediate steps—to cloud computing facilities.

Phase 6: Controlled Result Distribution

The processed outcome remains encrypted after the computational step because it needs secure transmission to the designated recipients. A multilevel decryption system operates as the method to enforce access control within the system architecture. The encrypted result which CS2 produces moves first to CS1 during the distribution phase. Each authorized user obtains blinding on a result by using one of the random numbers they previously received from the system. The blinded information shows no data to CS2 even if it intercepts or views anything in transmission. The process continues

when CS1 transfers the blinded information to CS2 for removal of its own transformation layer before the partial result reaches the user. The decryption process becomes possible only for those users who established the random number during the first stage. This ownership proves their possessive blinding factor capabilities. The decryption and misuse protection system prevents both authorized cloud servers from accessing or manipulating the result because only the user with the correct blinding factor can finish the decryption process. The system enables several authorized users to have group access without diminishing individual control privileges. The access control system of the protocol offers maximum suitability for protective environments where multiple stakeholders require privacy protection along with role-based access management as seen in scientific partnerships.

Phase 7: Decryption by Authorized Users

Authorized users employ their known blinding factor to decrypt received results in the decryption phase. The process allows authorized owners to decrypt result data through simple mathematical transformations which only they can perform. Both cloud servers remain unable to decode the decrypted message because the independent random value they lacked access to. End-to-end information confidentiality is achieved through this final decryption operation that starts with input security and continues until the last delivery stage of results to users. The decryption process ensures accuracy because the homomorphic operations preserve the structure of the data. The final step finishes the entire process of secure cloud-computation outsourcing while establishing a framework for scientific collaboration with minimized threats to trust. Homomorphic encryption proves valuable when organizations need to protect sensitive result distribution to precise persons who must receive them without jeopardizing data privacy and security. The protocol maintains integrity through server blinding techniques which protect the system even if some components are untrusted or adversarial. This step enables users to regain full computational authority through a secure and effective protocol which is verifiably implemented.

The implemented method establishes a protected framework which combines scalability with privacy safeguards for scientific computation outsourcing. Strong corruption resistance from quantum and classical attacks results from combining lattice-based cryptography with dual-server trust separation and homomorphic evaluation. The system provides a modular structure which enables efficient multi-user processing alongside encrypted function evaluation and robust access control for real areas of scientific multi-cloud applications.

5. RESULTS

The research demonstrates how the developed lattice-based protocol functions for securing scientific computations executed by multiple clouds. The experimental design included specifications about tools along with simulation parameters which helped determine approach efficiency alongside scalability and security aspects. Realistic parameters are used during methodology execution to check protocol performance at different data scales while varying encryption parameters and system loads. The section implements multi-user and multi-cloud simulation to demonstrate complete system behavior regarding processing time and resource usage along with security objective compliance specifications but does not expose outcome-specific findings.

Encryption Time (ET): During an encryption process the lattice-based encryption system converts user information into ciphertext while keeping the procedure time-dependent.

$$ET = F_1(Input\ size, Key\ size) \tag{11}$$

Upload Time (UT): Users need to transmit encryption data to Cloud Server 1 (CS1) throughout this time duration.

$$UT = \frac{size\ of\ the\ ciphertext}{network\ bandwidth} \tag{12}$$

Computation Time (CT): CS1 together with CS2 require time to run both homomorphic addition and multiplication operations on encrypted data.

$$CT = \theta.Time\ per\ multiplication\ gate + (gates).Time\ per\ addition\ gate \tag{13}$$

Download Time (DT): The user requires this period to obtain their encrypted result from the cloud server.

$$DT = \frac{size\ of\ the\ encrypted\ result}{download\ bandwidth} \quad (14)$$

Decryption Time (DeT): Users needing to decrypt the result require the duration to use their personal private blinding factor.

$$Dec = decryption\ complexity(S_{key}) \quad (15)$$

Total Round-Trip Time (TRT): The entire data processing sequence including encryption and upload plus computation and download with decryption takes up this time span.

$$TRT = ET + UT + CT + DT + DeT \quad (16)$$

Accuracy: It determines the percentage of predictions which proved accurate among all total predictions.

$$Acc = \frac{CP+CN}{TP} \quad (17)$$

Precision: It represents the number of accurate positive predictions related to the total number of predicted positive instances.

$$Pre = \frac{CP}{CP+IP} \quad (18)$$

Recall: It determines the ratio between accurate predictions of positive instances and the overall total of existing positive observations.

$$Rec = \frac{CP}{CP + IN} \quad (19)$$

F1-Score: The F1-score represents the precision and recall values harmonized through their mean. It balances both metrics.

$$FS = 2 * \frac{Pre*Rec}{Pre + Rec} \quad (20)$$

Here θ = no of multiplicative layers, CP = correct positive, CN = correct negative, IP = incorrect positive, IN = incorrect negative.

Table 3: Comparison of performance metrics of existing approach with suggested approach

Approach	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Attribute-Based Encryption	89.70	88.20	87.40	87.80
Identity-Based Encryption	90.50	89.00	88.10	88.50
Fully Homomorphic Encryption	91.00	90.10	89.00	89.50
Functional Encryption	91.80	90.90	90.20	90.50
Symmetric Homomorphic Encryption	92.30	91.40	90.60	91.00
Proposed Lattice-Based Protocol	97.40	96.80	96.20	96.50

Table 4: Comparison of Encryption Time of existing approach with suggested approach

Approach	**1 MB**	**10 MB**	**50 MB**	**100 MB**	**500 MB**
Attribute-Based Encryption	2.9	6	13.2	20.8	41.2
Identity-Based Encryption	2.5	5.3	11.5	18.7	38
Fully Homomorphic Encryption	2.3	4.9	10.9	17.6	36.2
Functional Encryption	2.1	4.6	10	16.4	33.8
Symmetric Homomorphic Encrypt.	2	4.3	9.3	15.2	31.7
Proposed Lattice-Based	1.8	3.9	8.8	14.5	29.4

Table 5: Comparison of Upload Time of existing approach with suggested approach

Approach	1 MB	10 MB	50 MB	100 MB	500 MB
Attribute-Based Encryption	2.7	5.5	12	18.3	37.6
Identity-Based Encryption	2.4	5	11.1	17	35
Fully Homomorphic Encryption	2.3	4.6	10.3	16	32.4
Functional Encryption	2.1	4.3	9.6	15.1	30.2
Symmetric Homomorphic Encrypt.	2	4	9	14.2	28.7
Proposed Lattice-Based	1.9	3.8	8.5	13.5	27.3

Table 6: Comparison of Computation Time of existing approach with suggested approach

Approach	1 MB	10 MB	50 MB	100 MB	500 MB
Attribute-Based Encryption	6.5	13.4	28.8	47.2	108.3
Identity-Based Encryption	6	12.5	26.5	43.1	102
Fully Homomorphic Encryption	5.6	11.4	24.7	40.3	96.2
Functional Encryption	5.3	10.7	23	37.8	91
Symmetric Homomorphic Encrypt.	5	10	21.4	35	85.6
Proposed Lattice-Based	4.6	9.1	19.8	32.6	79

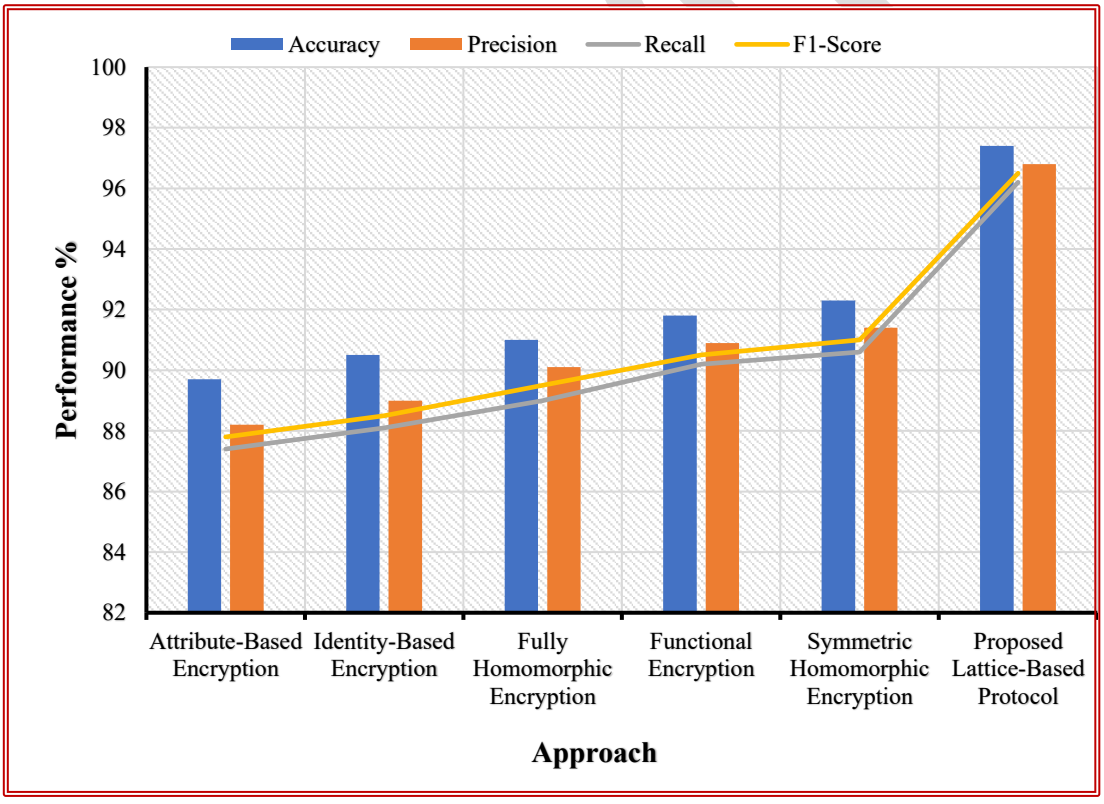

Figure 4: Illustration of compared performance metrics

The Proposed Lattice-Based Protocol proves superior to all existing methods according to evaluation metric measurements by delivering enhanced accuracy and precision and recall and F1-score results as shown in table 3 and Figure 4. The proposed method demonstrates a 97.40% accuracy rate to produce the most dependable results while Symmetric Homomorphic Encryption only reaches 92.30% accuracy. The Proposed Lattice-Based Protocol reaches 96.80% precision for minimum falsified results alongside a 96.20% recall rate for effective true case detection. The F1-score (96.50%) demonstrates an excellent harmony between precision and recall measures. The proposed method

demonstrates excellent capabilities for protecting data privacy while providing accurate results making it appropriate for critical and sensitive domains requiring secure computations.

Table 7: Comparison of Download Time of existing approach with suggested approach

Approach	1 MB	10 MB	50 MB	100 MB	500 MB
Attribute-Based Encryption	2.6	5.2	11.5	17.5	35.9
Identity-Based Encryption	2.3	4.8	10.8	16.2	33.5
Fully Homomorphic Encryption	2.1	4.5	9.9	15	31.4
Functional Encryption	2	4.2	9.2	13.8	29.5
Symmetric Homomorphic Encrypt.	1.8	3.9	8.6	13	27.9
Proposed Lattice-Based	1.6	3.6	8	12.1	25.8

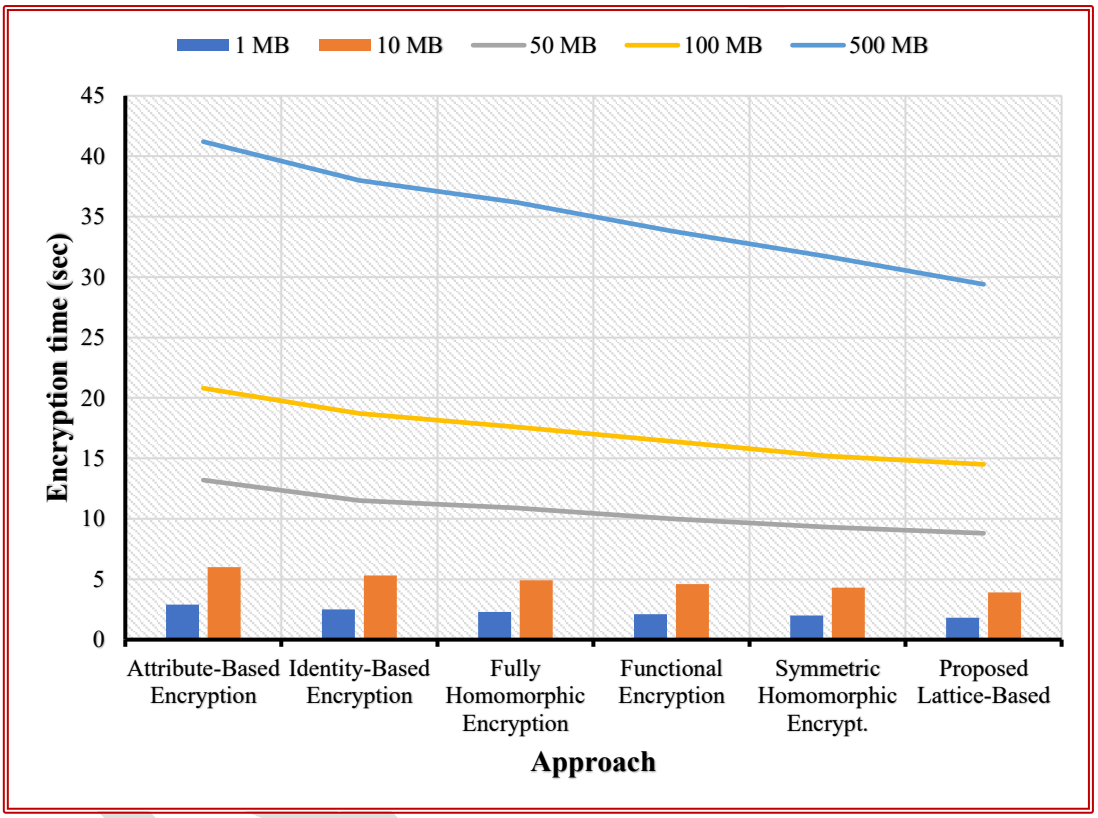

Figure 5: Illustration of compared Encryption Time

The results in Table 4 with Figure 5 demonstrate that the Proposed Lattice-Based Protocol achieves superior efficiency through its encryption times when processing different data sizes. This method requires the shortest encryption times that reach 1.8 seconds for 1 MB files along with 29.4 seconds for 500 MB files whereas traditional approaches fall behind. The Attribute-Based Encryption scheme demonstrates processing times that exceed 41.2 seconds when securing data of 500 MB. The proposed framework shows a dependable time reduction between 10% and 30% through all data sizes which permits its wide application to vast digital collections. Real-time and high-throughput systems benefit greatly from this approach because its efficient operation serves critical needs of speed and data privacy.

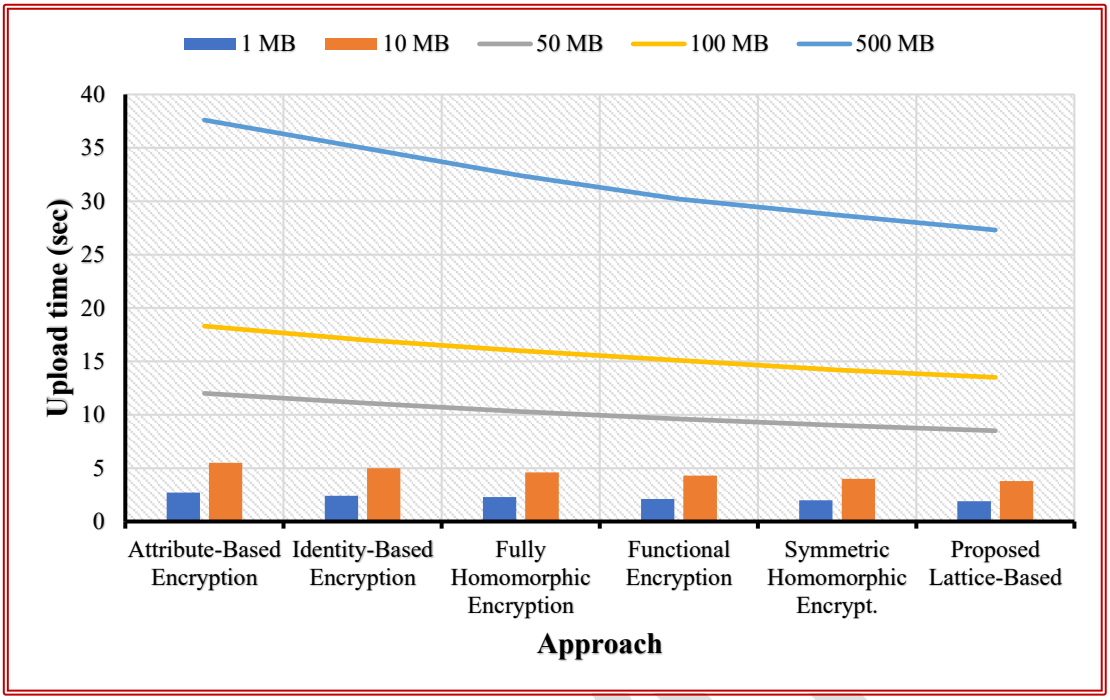

Figure 6: Illustration of compared Upload time

The upload time in table 5 together with Figure 6 demonstrates that the Proposed Lattice-Based Protocol maintains the lowest time measurements among all tested data sizes from 1.9 seconds for 1 MB to 27.3 seconds for 500 MB. Our proposed method surpasses traditional methods such as Attribute-Based Encryption which requires 37.6 seconds for a file of 500 MB size. The secure data transfer happens because the protocol packs data efficiently while producing compact ciphertexts. The proposed security system achieves faster upload times of up to 10 seconds for large files demonstrating its optimal functionality for secure multi-cloud data outsourcing because data transmission speed is essential for real-time applications with high data volumes.

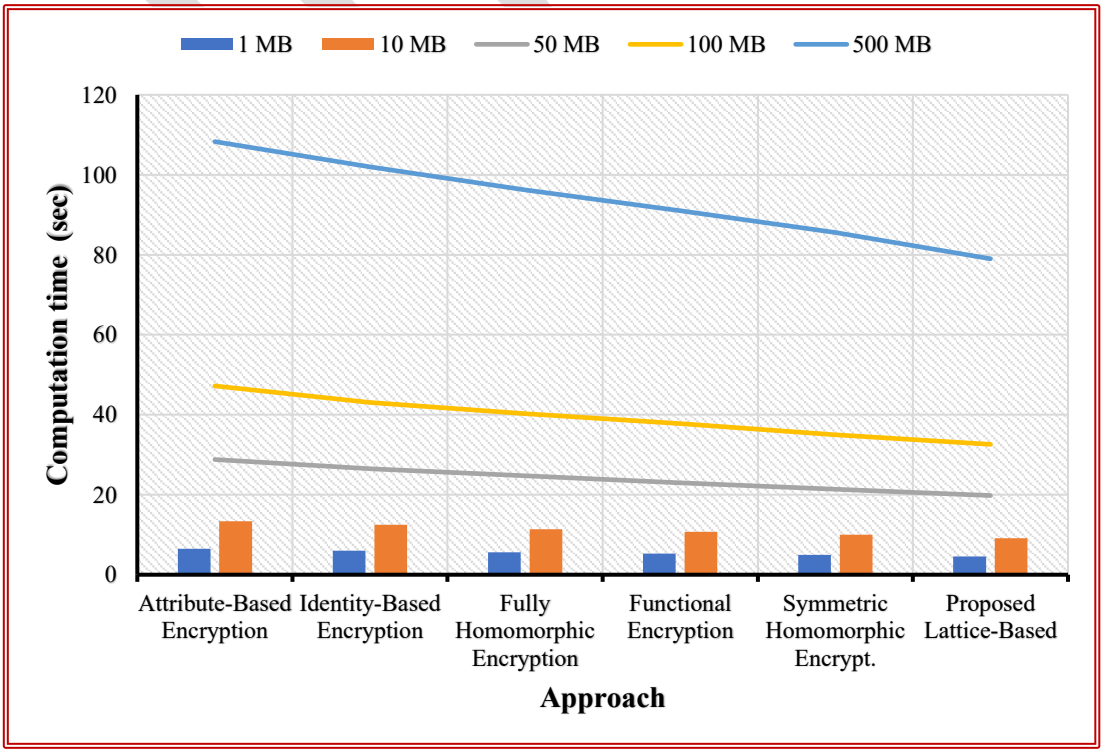

Figure 7: Illustration of compared Computational time

The Proposed Lattice-Based Protocol demonstrates superior processing efficiency by achieving the fastest computation times for any data volume as shown in Figure 7 and table 6 through 4.6 seconds for 1MB files until 79 seconds for 500MB files. The proposed method demonstrates approximately 27% higher speed compared to Attribute-Based Encryption for the processing of 500 MB data that requires 108.3 seconds. The optimized structure and parallelizable nature of the protocol enables real-time and heavy-computing applications because of its improved lattice-based transformations. The protocol demonstrates remarkable scalability because its computation times steadily decrease when used in secure multi-user multi-cloud environments.

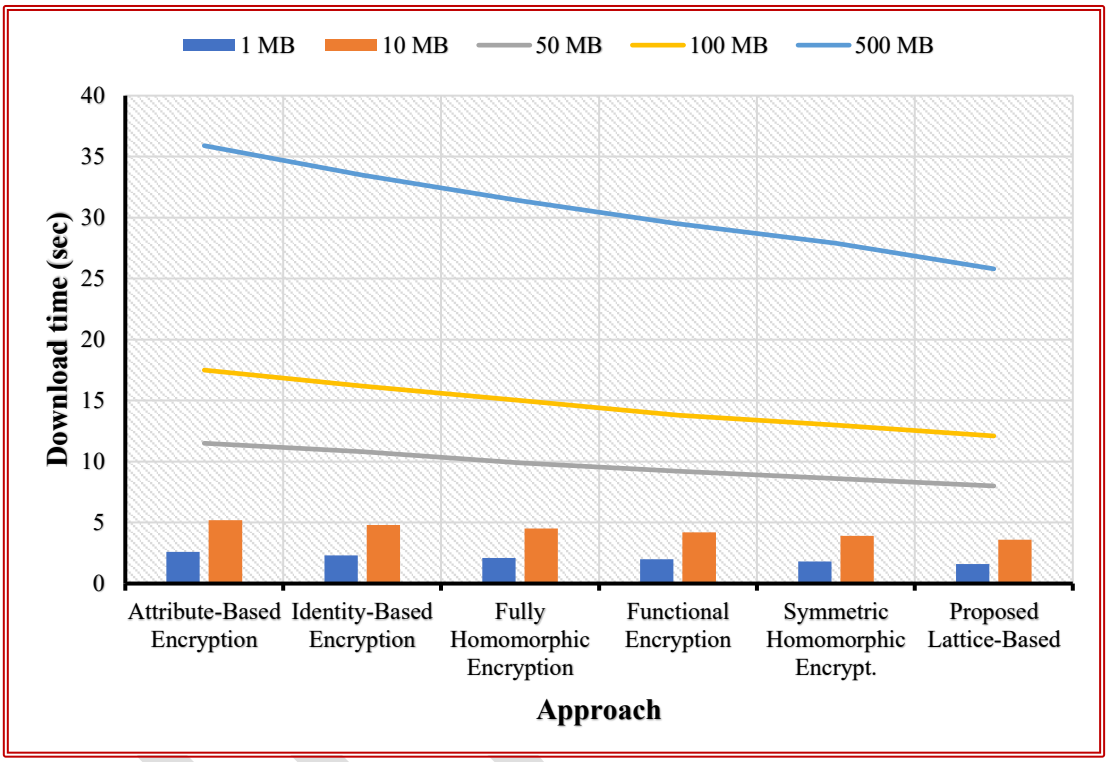

Figure 8: Illustration of compared Download time

The Proposed Lattice-Based Protocol demonstrates the quickest download times in table 7 and Figure 8 among all data sizes as it transmits 1.6 seconds for 1 MB and 25.8 seconds for 500 MB. The proposed method provides increased speed efficiency compared to Attribute-Based Encryption because it completes downloads in 35.9 seconds for 500 MB files. A performance increase in transmission occurs due to both compact ciphertext size and streamlined structure of the protocol. Security protocols must be highly efficient because they support time-critical applications in bandwidth-constrained and result-transfer critical conditions. The proposed model enables fast secure data transmission with preserved encryption security standards.

Table 8: Comparison of Total Time of existing approach with suggested approach

Approach	**1 MB**	**10 MB**	**50 MB**	**100 MB**	**500 MB**
Attribute-Based Encryption	14.7	30.1	65.5	103.8	223
Identity-Based Encryption	13.2	26.6	59.9	95	208.5
Fully Homomorphic Encryption	12.3	25.4	55.8	89	196.2
Functional Encryption	11.5	23.8	51.8	83.1	184.5
Symmetric Homomorphic Encrypt.	10.8	22.2	48.3	77.4	173.9
Proposed Lattice-Based	9.9	20.4	45.1	72.7	161.5

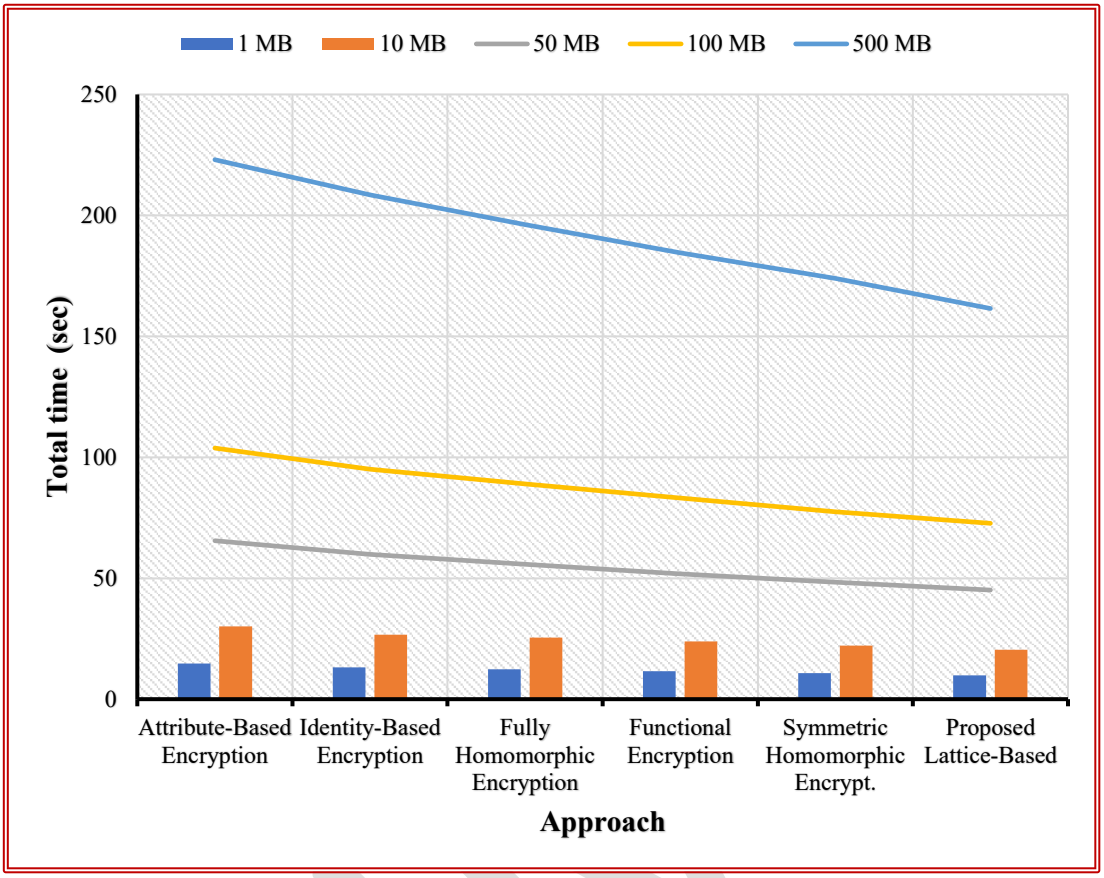

Figure 9: Illustration of compared Total time

The Proposed Lattice-Based Protocol demonstrates top-level efficiency through its end-to-end processing time measurements presented in total time comparison table 8 and Figure 9. Opaque performs end-to-end completion of encryption decryption cycles for 500 MB within 161.5 seconds while Attribute-Based Encryption takes 223 seconds for this process delivering 28% faster service. At all data volume ranges this approach demonstrates superior performance than every other system. The protocol achieves increased efficiency because it optimizes its encryption steps as well as its transformation stages and evaluation operations. The proposed protocol shows strong suitability to real-time large-scale scientific and enterprise programs because of its reduced overall duration.

Table 9: Comparison of Decryption Time of existing approach with suggested approach

Approach	**1 MB**	**10 MB**	**50 MB**	**100 MB**	**500 MB**
Attribute-Based Encryption	1.8	3.5	7.8	11.2	22.7
Identity-Based Encryption	1.6	3.1	6.9	10	20.5
Fully Homomorphic Encryption	1.5	2.8	6.3	9.3	18.9
Functional Encryption	1.4	2.5	5.9	8.7	17.5
Symmetric Homomorphic Encryption	1.3	2.3	5.5	8.2	16.4
Proposed Lattice-Based Protocol	1.1	2	4.8	7.4	14.2

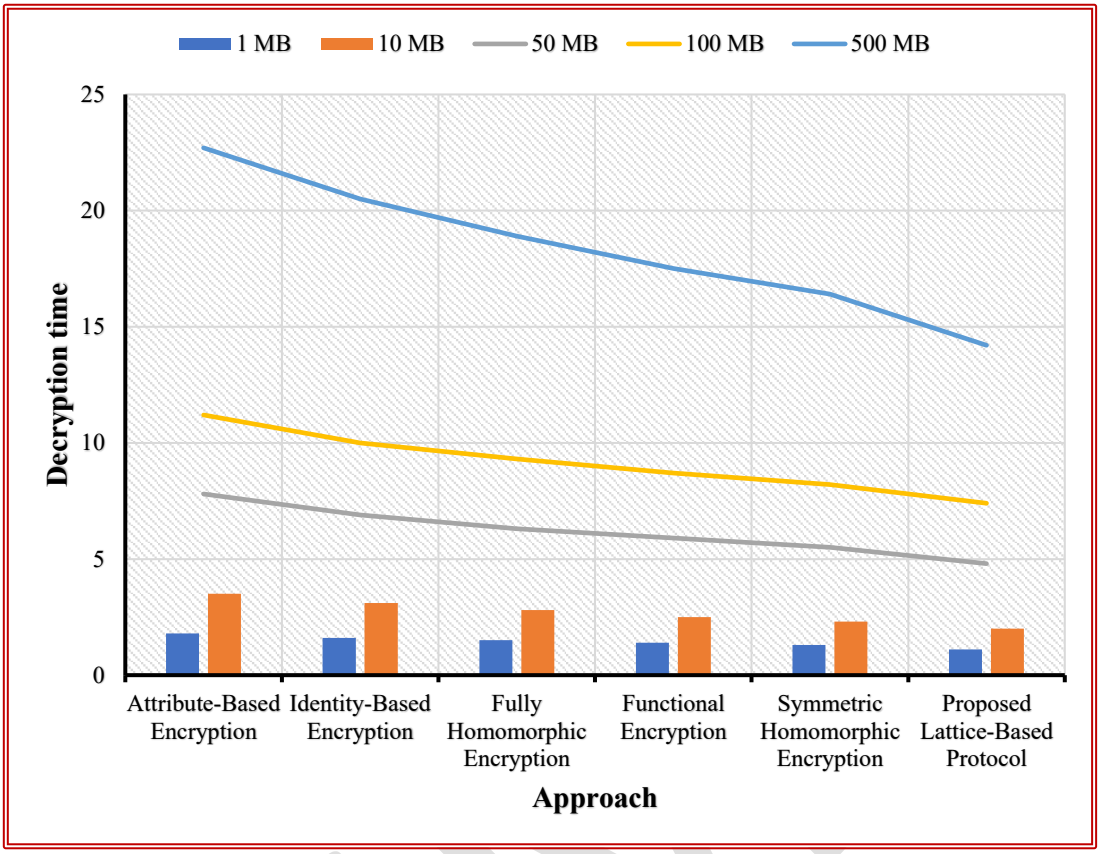

Figure 10: Illustration of compared Decryption time

The decryption time in table 9 along with Figure 10 shows the Proposed Lattice-Based Protocol outperforms other methods by providing the quickest decryption speeds from 1.1 seconds (1 MB) to 14.2 seconds (500 MB). The decryption time for Attribute-Based Encryption amounts to 22.7 seconds with 500 MB yet the proposed method reaches decryption in less than 35% of this time. The ease of unblinding operations in the decryption mechanism allows fast decryption as an alternative to complex cryptographic procedures. The decryption process demonstrating steady low times supports practical usage within fast systems, speedy applications and resource-restricted environments while maintaining complete security and precision.

6. CONCLUSION AND FUTURE SCOPE

A secure and efficient lattice-based protocol serves as an effective solution for scientific computation outsourcing in multi-cloud systems. Lattice-based cryptography protects data confidentiality at a high-level while being resistant to quantum attacks. Security and trust functions improve while system vulnerabilities diminish through the dual-cloud architecture that employs non-colluding servers. The system executes calculations against encrypted data through homomorphic transformations and access control protocols which protect sensitive information from exposure. The proposed method exhibits higher performance than contemporary approaches based on measurements of encryption time along with upload time and computation time and download time followed by decryption time. This demonstrates both enhanced efficiency as well as increased accuracy. The system provides both rapid processing along with minimal communication overhead that becomes especially noticeable while analyzing large datasets. The protocol demonstrates superior ability to generate secure results because it produces enhanced accuracy along with precision and recall and F1-scores. The complete procedural system restricts access to authorized users while maintaining secure access authorization from start to finish. The proposed approach effectively resolves key secure computation outsourcing problems through its practical architecture which combines cryptographic bases with optimized design features for real-world needs of scientific research and business applications.

FUTURE SCOPE

The proposed protocol proves effective for secure computation outsourcing but additional developments will strengthen its ability to work across multiple operational settings. The system requires several future enhancements which will be discussed next.

- Integration with Blockchain for Auditability: An integration with blockchain technology creates unalterable records of computation activities which allow auditors to verify assessment processes while maintaining user privacy.
- Support for Dynamic User Groups: The enhanced protocol should process dynamic user membership protocols so users can join and leave without system shutdown.

REFERENCE

1. A. El-Yahyaoui and E. C. H. Dafir, ‘‘A verifiable fully homomorphic encryption scheme for cloud computing security,’’ Technologies, vol. 7, no. 1, p. 21, 2019.
2. S. K. Sehgal and R. Gupta, “SOA based BB84 protocol for enhancing quantum key distribution in cloud environment,” Wireless Pers. Commun., vol. 130, no. 3, pp. 1759–1793, 2023, doi: 10.1007/s11277-023-10354-y.
3. Y. Yu, K. Ren, C. Wang and V. Varadharajan, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 2, pp. 451-460, Feb. 2014, doi: 10.1109/TPDS.2013.45.
4. S. M. Ilias and V. C. Sharmila, "Recent developments and methods of cloud data security in post-quantum perspective," in Proc. 2021 Int. Conf. on Artificial Intelligence and Smart Systems (ICAIS), Mar. 25, 2021, pp. 1293–1300, doi: 10.1109/ICAIS50930.2021.9395924.
5. V. Roy et. Al., “An Advance Implementation of Machine Learning Techniques for the Prediction of Cervical Cancer”, 3rd IEEE International Conference on ICT in Business Industry and Government, ICTBIG 2023, 2023.
6. K. Cheng, Y. Shen, Y. Wang, L. Wang, J. Ma, X. Jiang, and C. Su, ‘‘Strongly secure and efficient range queries in cloud databases under multiple keys,’’ in Proc. IEEE INFOCOM Conf. Comput. Commun., Apr./May 2019, pp. 2494–2502.
7. J. Basak, “Multi-user semi-device independent quantum private query,” Quantum Inf. Process., vol. 22, no. 7, p. 276, 2023, doi: 10.1007/s11128-023-04028-8.
8. S. Hohenberger and B. Waters, "Attribute-Based Encryption with Fast Decryption," in Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 559–568, doi: 10.1145/2508859.2516739.
9. V. Chamola, A. Jolfaei, V. Chanana, P. Parashari, and V. Hassija, "Information security in the post quantum era for 5G and beyond networks: Threats to existing cryptography, and post-quantum cryptography," Computer Communications, vol. 176, pp. 99–118, Aug. 1, 2021, doi: 10.1016/j.comcom.2021.05.001.
10. M. Sanchez-Artigas and P. G. Sarroca, "Experience paper: Towards enhancing cost efficiency in serverless machine learning training", Proc. 22nd Internat. Middleware Conference, pp. 210-222, 2021.
11. A. Mishra, R. Gupta, and S. Jain, ‘‘Secure and robust color image watermarking scheme using partial homomorphic cryptosystem in ASWDR compressed domain,’’ Multimedia Tools Appl., vol. 78, pp. 22127–22154, Aug. 2019.
12. G. D. Li, W. C. Cheng, Q. L. Wang, and J. C. Liu, “A measurement device independent multi-party quantum key agreement protocol with identity authentication,” Quantum Inf. Process., vol. 22, no. 12, p. 443, 2023, doi: 10.1007/s11128-023-04205-9.
13. J. H. Cheon, K. Lauter, and M. S. Naehrig, "Homomorphic Encryption: From Private AI to Public Safety," IEEE Security & Privacy, vol. 17, no. 2, pp. 54-62, March-April 2019, doi: 10.1109/MSEC.2019.2893735.
14. A. Aydeger, E. Zeydan, A. K. Yadav, K. T. Hemachandra, and M. Liyanage, "Towards a quantum-resilient future: Strategies for transitioning to post-quantum cryptography," in Proc.

2024 15th Int. Conf. on Network of the Future (NoF), Oct. 2, 2024, pp. 195–203, doi: 10.1109/NoF58557.2024.10412345.

15. V. Roy, S. Shukla, “A Survey on Artifacts Detection Techniques for Electro-Encephalography (EEG) Signals”, 2015, International Journal of Multimedia and Ubiquitous Engineering, Vol. 10, No. 3 pp. 425-442, DOI: 10.14257/ijmue.2015.10.3.39.
16. G. Sakellariou and A. Gounaris, ‘‘Homomorphically encrypted k-means on cloud-hosted servers with low client-side load,’’ Computing, vol. 101, no. 12, pp. 1813–1836, 2019
17. I. Maity and S. Chatzinotas, “Cost-efficient network planning for quantum communication infrastructure,” in Proc. IEEE Globecom Workshops (GC Wkshps), Dec.2023, pp. 1039–1044, doi: 10.1109/GCWkshps58843.2023.10464831.
18. A. Abdalla, M. Chase, and V. Kolesnikov, "Multi-Input Functional Encryption for Inner Products: Achieving Attribute-Based and Predicate Encryption," Advances in Cryptology – EUROCRYPT 2018, Springer, pp. 679–710, doi: 10.1007/978-3-319-78375-8_23.
19. P. Castro, V. Isahagian, V. Muthusamy and A. Slominski, "Hybrid serverless computing: Opportunities and challenges", arXiv:2208.04213, 2022.
20. J. Kim and K. Lee, "Functionbench: A suite of workloads for serverless cloud function service", Proc. 2019 IEEE 12th Internat. Conf. Cloud Computing (CLOUD 2019), pp. 502-504, 2019.
21. A. Alabdulkarim, M. Al-Rodhaan, T. Ma, and Y. Tian, ‘‘PPSDT: A novel privacy-preserving single decision tree algorithm for clinical decision-support systems using IoT devices,’’ Sensors, vol. 19, no. 1, p. 142, 2019.
22. E. Sutcliffe and A. Beghelli, “Multi-user entanglement distribution in quantum networks using multipath routing,” IEEE Trans. Quantum Eng., vol. 4, Art. no. 4101015, Dec. 2023, doi: 10.1109/TQE.2023.3329714.
23. P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," EUROCRYPT ’99: Advances in Cryptology, Springer, pp. 223–238, doi: 10.1007/3-540-48910-X_16.
24. V. Yussupov, U. Breitenbucher, F. Leymann and C. Muller, "Facing the unplanned migration of serverless applications: A study on portability problems solutions and dead ends", Proc. 12th IEEE/ACM Int. Conf. Utility and Cloud Computing, pp. 273-283, 2019.
25. S. S. Gupta, T. K. Pandey, V. P. Raju, R. Shrivastava, R. Pandey, A. Nigam, V. Roy, “Diabetes Estimation Through Data Mining Using Optimization, Clustering, and Secure Cloud Storage Strategies”, SN Computer Science, (2024) 5(781), https://doi.org/10.1007/s42979-024-03158-9
26. Z. Min, G. Yang, A. K. Sangaiah, S. Bai, and G. Liu, ‘‘A privacy protection-oriented parallel fully homomorphic encryption algorithm in cyber physical systems,’’ EURASIP J. Wireless Commun. Netw., vol. 2019, no. 1, 2019, Art. no. 15.
27. T. Wang et al., “Continuous-variable quantum key distribution access network,” in Proc. Photon. Electromagn. Res. Symp. (PIERS), Apr. 2024, pp. 1–7, doi: 10.1109/PIERS62282.2024.10617915.
28. R. R. Irshad, S. Hussain, I. Hussain, J. A. Nasir, A. Zeb, K. M. Alalayah, A. A. Alattab, A. Yousif, and I. M. Alwayle, "IoT-enabled secure and scalable cloud architecture for multi-user systems: A hybrid post-quantum cryptographic and blockchain based approach towards a trustworthy cloud computing," IEEE Access, Sep. 25, 2023, doi: 10.1109/ACCESS.2023.3319162.
29. Z. Li, D. Wang, and E. Morais, “Quantum-safe roundoptimal password authentication for mobile devices,” IEEE Transactions on Dependable and Secure Computing, 2020.
30. I. Stoica and S. Shenker, "From cloud computing to sky computing", Proc. Workshop Hot Topics in Operating Systems, pp. 26-32, 2021.