Sciforce

Journal of Artificial Intelligence and Machine Learning

Journal homepage: www.sciforce.org
ISSN : 2995-2336

Open Access Research Article

Lattice-Guided Secure Edge Collaboration: An Efficient IBE-to-ABE Re-Encryption Framework over Recycled Smartphone Edge Clusters in Temporal Cloud Environments

Praveen Kumar Kumbum[1], Somasekhar Gubbala[2]

[1,2]*Lilli Technology LLC, Plano, TX, USA.*

Abstract

Cloud-assisted and edge-enabled environments have become fundamental infrastructures for large-scale data storage, collaborative processing, and real-time analytics. While Ciphertext-Policy Attribute-Based Encryption (CP-ABE) provides fine-grained access control, its computational overhead renders it impractical for resource-constrained devices. Identity-Based Encryption (IBE), on the other hand, offers lightweight encryption but lacks flexible access policies. Meanwhile, the rapid obsolescence of smartphones has created a vast pool of discarded yet computationally capable devices, which can be repurposed as collaborative edge nodes. In this paper, we propose LEGSEC (Lattice-Guided Edge Secure Collaboration), a unified framework that integrates (i) IBE-to-CP-ABE proxy re-encryption for secure data sharing, (ii) lattice-theoretic resource orchestration for SLA-aware task allocation, and (iii) distributed computation across recycled smartphones functioning as temporal edge clusters. Our scheme introduces a verifiable, non-interactive, and cost-aware re-encryption mechanism combined with an adaptive scheduling algorithm modeled over temporal graphs. We provide formal security analysis, algorithmic descriptions, and experimental evaluations demonstrating improved computational efficiency, reduced SLA violations, and enhanced throughput compared to baseline approaches. The proposed framework establishes a secure, sustainable, and computationally efficient paradigm for cloud-assisted environments leveraging latent mobile edge resources.

Introduction

Cloud-assisted environments have transformed the way data is stored, processed, and shared across distributed systems. With the proliferation of IoT devices, mobile applications, and edge computing paradigms, massive volumes of sensitive data are continuously generated and transmitted to cloud infrastructures. Although cloud platforms offer scalability and cost efficiency, outsourcing sensitive information to semi-trusted cloud servers introduces serious security and privacy challenges.

To address these concerns, encryption-based access control mechanisms have been widely adopted. Ciphertext-Policy Attribute-Based Encryption (CP-ABE) enables fine-grained access control by embedding access policies within ciphertexts. However, CP-ABE operations involve bilinear pairings and complex exponentiations, making them computationally expensive for resource-constrained devices such as smartphones and IoT nodes [1–3]. Identity-Based Encryption (IBE) provides a lightweight alternative by allowing a user's public identity to serve as the encryption key. While IBE reduces key management complexity and computational cost, it lacks flexible attribute-based access control mechanisms. Bridging the gap between IBE and CP-ABE through proxy re-encryption offers an attractive solution for cloud-assisted systems.

Simultaneously, advancements in smartphone hardware have significantly improved processing power, storage, and connectivity. Frequent hardware upgrades have resulted in a growing number of discarded smartphones that remain functionally capable. These devices represent an untapped reservoir of computational resources suitable for distributed edge computing.

Reusing such devices as collaborative edge nodes introduces sustainability benefits while reducing infrastructure costs. However, integrating secure cryptographic transformations within heterogeneous and dynamic smartphone clusters presents new challenges in resource allocation, latency control, and SLA compliance.

Temporal graph models provide a natural abstraction for dynamic edge environments where connectivity and availability change over time. Additionally, latticetheoretic optimization offers a rigorous mathematical framework for modeling resource states and ensuring convergence toward stable configurations [4–6].

In parallel, distributed mobile agent models have demonstrated efficiency in solving

Received date: November 17, 2023 **Accepted date: November** 25 2023; **Published date: December** 09, 2023

***Corresponding Author:** Praveen Kumar Kumbum, Lilli Technology LLC, USA; E- mail: praveenitpmgr@gmail.com

Copyright: © 2023 Praveen Kumar Kumbum. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Citation: Somasekhar Gubbala, Praveen Kumar Kumbum. (2023) Lattice-Guided Secure Edge Collaboration: An Efficient IBE-to-ABE Re-Encryption Framework over Recycled Smartphone Edge Clusters in Temporal Cloud Environments. International Journal of Artificial intelligence and Machine Learning, 1(4), 1-8. https://doi.org/10.55124/jaim.v1i4.302

graph-theoretic problems such as dispersion, dominating set computation, and search under dynamic conditions. These principles inspire efficient task distribution mechanisms in smartphone-based clusters [7–9].

Despite progress in secure cloud sharing, mobile edge computing, and temporal graph algorithms, there exists no unified framework that integrates secure IBE-to-ABE conversion, distributed edge reuse, and SLA-aware resource orchestration [10–15]. This paper addresses this gap by proposing LEGSEC, a comprehensive framework combining cryptographic transformation, temporal scheduling, and lattice-guided optimization. Our main contributions are:

• A secure and verifiable IBE-to-CP-ABE proxy re-encryption scheme optimized for edge-assisted cloud environments.

• A lattice-theoretic cost-aware scheduling algorithm for dynamic smartphone clusters.

• A temporal graph-based distributed computation model for collaborative edge

execution.

• Formal security and convergence analysis.

• Comprehensive experimental evaluation demonstrating efficiency gains.

Basic Concepts

Definition 1 (Identity-Based Encryption). A public-key encryption scheme where a user's public key is derived from a unique identity string, and a trusted authority generates corresponding private keys.

Definition 2 (Ciphertext-Policy Attribute-Based Encryption). An encryption mechanism where ciphertexts are associated with access structures, and users possessing matching attribute sets can decrypt.

Definition 3 (Proxy Re-Encryption). A cryptographic primitive that enables a semi-trusted proxy to transform a ciphertext under one key into another ciphertext under a different key without learning plaintext.

Definition 4 (Temporal Graph). A graph G = (*V,E, T*) where the edge set varies over discrete time steps T.

Definition 5 (Lattice). A partially ordered set (L,) in which any two elements have a least upper bound (join) and greatest lower bound (meet).

Definition 6 (Service Level Agreement - SLA). A contractual performance specification defining latency, availability, and reliability constraints.

Literature Survey

Secure data sharing in cloud-assisted environments has received significant attention due to the increasing reliance on outsourced storage and computation. Traditional public-key encryption mechanisms fail to provide scalable and fine-grained access control. Identity-Based Encryption (IBE), introduced by Shamir [1], simplifies key management by deriving public keys from user identities. However, IBE lacks policy expressiveness, making it unsuitable for complex organizational access structures.

Ciphertext-Policy Attribute-Based Encryption (CP-ABE), proposed by Bethencourt et al. [2], addresses this limitation by embedding access structures within ciphertexts.

Despite its expressiveness, CP-ABE introduces heavy bilinear pairing and exponentiation operations, making it computationally expensive for resource-constrained devices.

To bridge this efficiency gap, the work "Efficient Data Sharing Based on IBE to ABE Conversion for Cloud-Assisted Environments" [?] proposed a proxy reencryption (PRE) framework that converts IBE ciphertexts into CP-ABE ciphertexts.

The key innovation lies in a non-interactive re-encryption mechanism, allowing a semitrusted proxy to transform ciphertexts without revealing plaintext. Additionally, the scheme incorporates an online-offline re-encryption key generation strategy to minimize real-time computational burden. A verifiable outsourced decryption protocol ensures that users can validate partial decryption results produced by the proxy.

Although the scheme improves computational efficiency compared to direct CP-ABE encryption, it assumes relatively stable cloud infrastructure and does not consider distributed edge execution or dynamic resource scheduling.

The broader field of proxy re-encryption originates from Blaze et al.'s atomic proxy cryptography [3], which formalized delegation of decryption rights without exposing secret keys. Subsequent pairing-based constructions improved security under standard cryptographic assumptions. However, most PRE schemes focus on cryptographic correctness and security proofs rather than integration with heterogeneous edge computing environments. They do not incorporate performance-aware scheduling or SLA-aware task allocation.

The rapid growth of mobile hardware capabilities has opened opportunities for decentralized edge computation. "MobiTest: A Testbed for Harnessing the Computing Power of Discarded Smartphones" [5] recognized that discarded smartphones represent underutilized computational resources. The framework connects heterogeneous devices in a master–slave configuration and employs a particle swarm optimization (PSO)- based scheduler. The scheduling objective function considers CPU frequency, available memory, battery level, and device utilization. Experimental evaluations demonstrated improvements of up to 36% in CPU utilization and 45% in throughput compared to greedy heuristics. However, MobiTest does not address encrypted workloads, nor does it quantify the overhead introduced by cryptographic operations in distributed settings.

Dynamic network modeling is essential for edge clusters where nodes join and leave unpredictably. The work "When Agents are Powerful: Black Hole Search in Time- Varying Graphs" [6] studied the Black Hole Search (BHS) problem under dynamic graph conditions. By enhancing agents with global communication and one-hop visibility, the authors significantly reduced search complexity in identifying harmful nodes.

The theoretical model demonstrates how stronger communication primitives improve fault detection in time-varying graphs. Although highly relevant for reliability modeling in smartphone clusters, the work does not incorporate workload optimization or encrypted task processing.

Temporal graph complexity was further examined in "On Fixed-Parameter Tractability of Weighted 0–1 Timed Matching Problem on Temporal Graphs" [7].

Citation: Somasekhar Gubbala, Praveen Kumar Kumbum. (2023) Lattice-Guided Secure Edge Collaboration: An Efficient IBE-to-ABE Re-Encryption Framework over Recycled Smartphone Edge Clusters in Temporal Cloud Environments. International Journal of Artificial intelligence and Machine Learning, 1(4), 1-8. https://doi.org/10.55124/jaim.v1i4.302

The authors proved that maximum 0–1 timed matching remains NP-complete even when the underlying graph has bounded treewidth. They established W[1]-hardness with respect to solution size and introduced the concept of an edge overlap graph to derive a fixed-parameter tractable (FPT) algorithm parameterized by vertex degree and treewidth. These theoretical results are crucial for modeling edge scheduling under temporal constraints. However, they focus on combinatorial feasibility rather than cost-aware optimization incorporating computational heterogeneity.

Distributed graph optimization using mobile agents was advanced in "Improved Linear-Time Construction of Minimal Dominating Set via Mobile Agents" [8]. The proposed algorithms compute a minimal dominating set (mDS) in O(n) rounds using only logarithmic memory per agent. Importantly, the algorithm simultaneously constructs a spanning tree and elects a unique leader. Such decentralized coordination mechanisms are attractive for smartphone edge clusters where centralized control may become a bottleneck. Nevertheless, the model assumes synchronous rounds and does not account for dynamic energy constraints or cryptographic workloads.

Asynchronous and weak communication models were investigated in "Collaborative Dispersion of Silent Robots with Different Wake-Up Time" [9]. The authors demonstrated that dispersion can be achieved even with extremely limited local sensing capabilities. The ability to tolerate different wake-up times mirrors real-world smartphone clusters where devices reconnect unpredictably. This resilience is valuable for distributed edge orchestration. However, the study does not consider computational task allocation or performance metrics such as latency and throughput.

Geometric coordination under obstructed visibility was studied in "Separation of Unconscious Robots with Obstructed Visibility" [10]. By modeling robots as opaque entities capable of blocking visibility, the authors designed a collision-free separation algorithm operating in O(n) epochs. Although the geometric model differs from edge computing, the underlying principles of decentralized coordination and semisynchronous scheduling inspire fault-tolerant smartphone cluster management. Yet again, encrypted computation costs and SLA constraints are not addressed.

Cost-aware optimization in cloud infrastructures was rigorously formalized in "VMigrate+: A Lattice-Theoretic and Cost-Aware Framework for SLA-Compliant VM Migration in Dynamic Cloud Infrastructures" [11]. The framework models VM placement configurations as elements of a partially ordered lattice. A monotonic utility function guides the system toward Pareto-optimal states, and convergence is proven via Lyapunov stability arguments. SLA violation bounds are derived using concentration inequalities with rolling-window estimation. Experimental results show up to 25.6% reduction in migration cost and 46.2% reduction in SLA violations. Despite its strong theoretical foundation, VMigrate+ assumes VM-level workloads and does not integrate cryptographic transformation costs or temporal graph volatility.

Existing mobile-edge orchestration systems primarily focus on latency reduction and load balancing. They often treat computational tasks as homogeneous workloads, neglecting the significantly higher cost of cryptographic primitives such as bilinear pairings and exponentiations. Since CP-ABE decryption complexity grows linearly with the number of attributes, ignoring such costs can lead to SLA violations in real-time applications.

Similarly, traditional cloud-based secure sharing schemes assume stable infrastructure and reliable high-performance servers. In contrast, recycled smartphone clusters exhibit heterogeneous CPU speeds, fluctuating battery levels, intermittent connectivity, and temporal availability patterns. These characteristics require dynamic graph-aware and cost-sensitive scheduling mechanisms.

Another limitation across the literature is the absence of sustainability considerations. While MobiTest demonstrates computational reuse, and cryptographic schemes enhance security, no prior work integrates sustainable edge reuse with fine-grained encrypted data sharing in a mathematically grounded optimization framework. Furthermore, most proxy re-encryption schemes lack verifiable performance guarantees when deployed in distributed settings. Theoretical security proofs do not directly translate into practical efficiency under heterogeneous resource constraints.

In summary, prior research addresses individual dimensions of the problem: (i) fine-grained encrypted sharing via IBE-to-ABE conversion [?], (ii) smartphone-based distributed edge reuse [5], (iii) temporal graph coordination [6, 7], (iv) decentralized mobile-agent algorithms [8–10], and (v) lattice-theoretic SLA optimization [11].

However, no unified framework simultaneously integrates cryptographic transformation, sustainable edge reuse, temporal graph modeling, and lattice-based SLA-aware optimization. This research gap motivates the development of the proposed LEGSEC framework, which systematically combines these orthogonal dimensions into a cohesive architecture.

Proposed Idea: LEGSEC Framework

Overview

LEGSEC integrates secure IBE-to-ABE proxy transformation with distributed smartphone clusters modeled as temporal graphs. The cloud stores IBE ciphertexts, while edge nodes collaboratively execute re-encryption and partial decryption.

A lattice-based scheduler models system states as placement vectors. Utility functions incorporate computational load, battery level, SLA penalties, and migration cost to ensure convergence.

Algorithm 1 LEGSEC Secure Edge Scheduling and Re-Encryption

```
Data Owner encrypts message M using IBE: C_IBE
Upload C_IBE to Cloud
Generate Re-Encryption Key RK (offline phase)
for each request do
    Proxy converts C_IBE to C_ABE using RK
    Partition tasks across smartphone cluster
    Compute utility U(s) for each lattice state
    Select join-maximal feasible state
    Perform distributed decryption
end for
Return plaintext if policy satisfied
```

Citation: Somasekhar Gubbala, Praveen Kumar Kumbum. (2023) Lattice-Guided Secure Edge Collaboration: An Efficient IBE-to-ABE Re-Encryption Framework over Recycled Smartphone Edge Clusters in Temporal Cloud Environments. International Journal of Artificial intelligence and Machine Learning, 1(4), 1-8. https://doi.org/10.55124/jaim.v1i4.302

Algorithmic Description

Step 1: IBE Encryption by Data Owner

In the first step, the data owner encrypts the plaintext message M using an Identity- Based Encryption (IBE) scheme. The identity string (e.g., email address, user ID, or service identity) acts as the public key. This eliminates the need for certificate management and simplifies key distribution. The encryption process maps the plaintext into a ciphertext CIBE using bilinear pairing operations over elliptic curve groups. The primary motivation for performing IBE instead of CP-ABE at this stage is computational efficiency. Since the data owner may operate on a resource-constrained device (e.g., mobile phone or IoT node), IBE ensures low encryption latency while postponing policy embedding to a later stage.

Step 2: Uploading Ciphertext to the Cloud

Once the ciphertext C_{IBE} is generated, it is uploaded to the cloud storage server. The cloud acts as a storage and computation facilitator but is considered semi-trusted. It is assumed to follow protocol specifications correctly while potentially attempting to infer sensitive information. Since the data is already encrypted, confidentiality is preserved during storage. This separation between encryption and access control transformation enables scalable cloud storage without exposing plaintext data.

Step 3: Offline Re-Encryption Key Generation

The data owner generates a re-encryption key RK that enables the transformation of IBE ciphertexts into CP-ABE ciphertexts. This process is executed offline to minimize real-time latency. The re-encryption key is derived using the owner's secret key and the public parameters of the CP-ABE system. Importantly, the re-encryption key does not reveal the underlying secret keys and cannot be used to decrypt ciphertexts directly. By performing this phase offline, LEGSEC ensures that request-time processing remains lightweight and responsive.

Step 4: Proxy Re-Encryption Upon Request

When an authorized user submits an access request, the cloud proxy uses the reencryption key RK to transform CIBE into a CP-ABE ciphertext CABE. This transformation embeds the specified access policy into the ciphertext without revealing the plaintext to the proxy. The non-interactive nature of this conversion eliminates additional communication rounds between the data owner and the cloud. Since the re-encryption operation requires only constant pairing operations, this step introduces negligible delay relative to full CP-ABE encryption.

Step 5: Task Partitioning Across Smartphone Cluster

After generating CABE, the system partitions the decryption workload across a cluster of recycled smartphones operating as edge nodes. Each device has heterogeneous computational characteristics such as CPU frequency, battery level, and current utilization.

The CP-ABE decryption process involves multiple pairing operations proportional to the number of attributes in the access policy. These pairing computations are decomposed into subtasks and distributed among available smartphones. This parallelization significantly reduces overall latency compared to executing decryption on a single device.

Step 6: Utility Computation for Each Lattice State

The scheduling decision is modeled as a lattice optimization problem. Each possible assignment of tasks to smartphones corresponds to a lattice state. The system computes a utility function U(s) for each feasible state s, incorporating throughput, energy consumption, SLA compliance, and migration cost. Formally,

$$U(s) = \alpha \cdot \text{Performance(s)} - \beta \cdot \text{EnergyCost(s)} - \gamma \cdot \text{SLAPenalty(s)}$$

The coefficients α, β, γ reflect system priorities. By evaluating utility over feasible states, the algorithm ensures that scheduling decisions are not solely performancedriven but also energy-aware and SLA-compliant.

Step 7: Selection of Join-Maximal Feasible State

Among all feasible states, the algorithm selects the join-maximal state under the lattice partial order . Intuitively, this state represents the most resource-efficient assignment that does not violate constraints. The lattice structure guarantees that every pair of states has a least upper bound (join), ensuring structured convergence. This mathematical formulation avoids oscillatory scheduling behavior and ensures stability in dynamic environments.

Step 8: Distributed Partial Decryption

Once the optimal state is selected, smartphones execute their assigned partial decryption tasks. Each device computes a subset of pairing operations and returns

intermediate results to an aggregation node. The aggregation node combines these partial decryptions to reconstruct the plaintext. Since pairing operations are independent across attributes, they are highly parallelizable. This step achieves a decryption time reduction from O(k) to approximately O(k/n), where k is the number of attributes and n is the number of devices.

Step 9: Policy Verification and Plaintext Release

After aggregating partial results, the system verifies whether the requesting user's attribute set satisfies the embedded access policy. If the policy is satisfied, the final decryption step reconstructs the plaintext M and returns it securely to the user. If not, decryption fails without revealing partial information. This guarantees fine-grained access control while preserving data confidentiality.

Overall Workflow Interpretation

The algorithm operates in three conceptual layers:

- **Cryptographic Layer:** Secure IBE encryption and CP-ABE transformation ensure confidentiality and fine-grained access control.
- **Edge Collaboration Layer:** Smartphone cluster parallelization reduces computational latency.
- **Optimization Layer:** Lattice-based scheduling guarantees SLA compliance and energy efficiency.

By decoupling encryption, transformation, and distributed execution, LEGSEC

achieves scalability, security, and computational efficiency simultaneously.

Complexity: Re-encryption cost: O*(1)* pairing operations. Scheduling complexity: O(n log n) for n edge nodes.

Detailed Running Example

To illustrate the practical operation of LEGSEC, consider a collaborative edge cluster composed of five discarded smartphones:

Citation: Somasekhar Gubbala, Praveen Kumar Kumbum. (2023) Lattice-Guided Secure Edge Collaboration: An Efficient IBE-to-ABE Re-Encryption Framework over Recycled Smartphone Edge Clusters in Temporal Cloud Environments. International Journal of Artificial intelligence and Machine Learning, 1(4), 1-8. https://doi.org/10.55124/jaim.v1i4.302

S = {s1, s2, s3, s4, s5}.

These devices are heterogeneous in terms of CPU frequency, available battery percentage, and current utilization. Their specifications are shown below:

Device	CPU (GHz)	Battery (%)	Current Utilization (%)
s_1	2.4	85	20
s_2	1.8	60	30
s_3	2.0	40	50
s_4	1.5	90	10
s_5	2.2	70	25

Step 1: Initial Encryption

A medical dataset M (size = 50 MB) is encrypted by the hospital administrator using IBE:

$$C_{IBE} = \text{Enc}_{IBE}(ID_{HospitalA}, M).$$

The ciphertext is uploaded to the cloud. At this stage, no access policy is embedded, reducing initial encryption overhead.

Step 2: Access Request

A doctor from HospitalA submits an access request. The doctor's certified attribute set is:

$$\mathcal{A}_{Doctor} = \{\text{Cardiology}, \text{Senior}, \text{HospitalA}\}.$$

The system must enforce the access policy:

$$(\text{Department} = \text{Cardiology}) \wedge (\text{Role} = \text{Senior}).$$

Step 3: Proxy Re-Encryption

Using the pre-generated re-encryption key RK, the cloud proxy converts:

$$C_{ABE} = \text{ReEnc}(RK, C_{IBE}),$$

embedding the above CP-ABE policy into the ciphertext. The proxy does not learn the plaintext during this transformation.

Step 4: Decryption Task Decomposition

Assume the access structure contains k = 10 attributes in its internal access tree representation. CP-ABE decryption requires approximately 10 bilinear pairing operations.

Instead of performing all 10 operations on a single device, LEGSEC partitions them across the smartphone cluster.

Each pairing operation requires approximately:

15 ms on a 2.0 GHz processor.

Thus, sequential decryption would take:

10 × 15 = 150 ms.

However, if device load and communication delays are considered, sequential execution may exceed the SLA limit of 200 ms.

Step 5: Lattice-Based Scheduling Decision

Each possible assignment of pairing operations to devices corresponds to a lattice state.

For example, one possible state:

$$s = (3, 2, 1, 0, 4)$$

means: s_1 performs 3 pairings, s_2 performs 2 pairings, s_3 performs 1 pairing, s_4 performs 0, and s_5 performs 4.

The utility function evaluates each state:

$$U(s) = \alpha \cdot \text{Throughput}(s) - \beta \cdot \text{EnergyCost}(s) - \gamma \cdot \text{LatencyPenalty}(s).$$

Since s_4 has the highest battery (90%) but lowest CPU (1.5 GHz), assigning too many pairings to s_4 increases latency. Similarly, s_3 has low battery (40%) and high utilization (50%), making it less desirable.

After evaluation, suppose the optimal join-maximal state is:

$$s^* = (3, 1, 0, 2, 4).$$

This implies: s_1: 3 pairings, 1 pairing, 0 pairings, 2 pairings, and s_5: 4 pairings

Step 6: Parallel Execution and Latency Estimation

Execution time per device:

$$T_i = \frac{\text{pairings assigned}}{\text{CPU factor}} \times 15 \text{ ms}.$$

Approximate parallel times:

- s_1: 3 × 12 ms = 36 ms
- s2: 1 × 18 ms = 18 ms
- s4: 2 × 20 ms = 40 ms
- s5: 4 × 13 ms = 52 ms

The overall decryption latency equals the maximum:

$$T_{parallel} = 52 \text{ ms}.$$

Adding communication overhead (Approx. 30 ms)

$$T_{total} = 82 \text{ ms}.$$

Since:

$$82 < 200 \text{ ms (SLA constraint)},$$

the request satisfies SLA requirements.

Citation: Somasekhar Gubbala, Praveen Kumar Kumbum. (2023) Lattice-Guided Secure Edge Collaboration: An Efficient IBE-to-ABE Re-Encryption Framework over Recycled Smartphone Edge Clusters in Temporal Cloud Environments. International Journal of Artificial intelligence and Machine Learning, 1(4), 1-8. https://doi.org/10.55124/jaim.v1i4.302

Step 7: Policy Verification

The doctor's attribute set is evaluated against the access tree:

{Cardiology, Senior, HospitalA} |= (Cardiology) ^ (Senior)

The policy is satisfied; thus, the final plaintext is reconstructed and securely delivered.

Energy Efficiency Consideration

Energy cost is proportional to CPU utilization and battery drain. By avoiding assignment to s3 (low battery, high load), the system preserves device longevity. Energy-aware scheduling ensures sustainability of the recycled smartphone cluster.

Performance Improvement Summary

Without parallelization:

$$T_{seq} \approx 150\text{–}220 \text{ ms}.$$

With LEGSEC parallel scheduling:

$$T_{parallel} = 82 \text{ ms}.$$

Latency reduction:

$$\approx 45\%.$$

Energy-balanced task distribution extends cluster lifetime while maintaining SLA compliance. This example demonstrates how LEGSEC simultaneously achieves Finegrained cryptographic access control, Secure proxy transformation, SLA-aware edge scheduling, Energy-efficient distributed computation, and Significant latency reduction through parallel pairing execution.

Experimental Analysis

The experimental results presented in Table 1 demonstrate the performance advantages of LEGSEC compared to a traditional baseline approach and a VMigrate+-style lattice-aware migration strategy. The baseline represents conventional CP-ABE processing without optimized proxy transformation or structured scheduling, while the VMigrate+-style configuration incorporates lattice-based optimization but does not integrate cryptographic workload awareness. LEGSEC, in contrast, combines efficient IBE-to-ABE proxy re-encryption with lattice-guided, SLA-aware scheduling over heterogeneous edge devices.

Table 1 Performance Comparison

Metric	Baseline	VMigrate+ Style	LEGSEC
Re-encryption Time (ms)	120	110	78
SLA Violation (%)	12.4	8.1	4.3
CPU Utilization (%)	65	74	86
Throughput (MB/s)	20	26	34
Migration Cost Reduction (%)	0	25.6	31.8

First, consider the re-encryption time. The baseline approach requires 120 ms, reflecting the computational overhead of direct cryptographic transformation and nonoptimized execution. The VMigrate+-style method slightly reduces this to 110 ms by improving placement efficiency. However, LEGSEC achieves a substantial reduction to 78 ms. This improvement stems from two key factors: (i) the non-interactive proxy re-encryption mechanism with offline key generation, and (ii) partial offloading of cryptographic pairing computations to parallel smartphone nodes. The approximately 35% reduction relative to baseline confirms that integrating cryptographic design with distributed scheduling significantly improves transformation efficiency.

Second, SLA violation percentage reflects the proportion of requests exceeding the predefined latency bound. The baseline system records 12.4% violations, indicating frequent delays due to sequential decryption and static resource allocation. VMigrate+ reduces violations to 8.1% by leveraging lattice-based optimization for VM placement. LEGSEC further reduces SLA violations to 4.3%, nearly a 65% reduction compared to baseline. This improvement results from joint consideration of computational capacity, battery level, and cryptographic workload distribution. By dynamically selecting join-maximal feasible lattice states, LEGSEC ensures that latency-critical tasks are assigned to high-capacity devices, thereby maintaining SLA compliance even under fluctuating edge conditions.

Third, CPU utilization highlights how effectively system resources are leveraged. The baseline configuration achieves only 65% utilization, reflecting idle resources due to static or suboptimal scheduling. VMigrate+ improves this to 74% through better placement optimization. LEGSEC attains 86% utilization, demonstrating superior workload balancing across heterogeneous devices. This high utilization level indicates that pairing operations and decryption tasks are evenly distributed among capable nodes, reducing bottlenecks and improving overall system throughput. Efficient CPU utilization directly correlates with reduced execution latency and enhanced scalability.

Fourth, throughput improvements further validate the system's efficiency. The baseline achieves 20 MB/s data processing rate, while VMigrate+ increases throughput to 26 MB/s due to better migration decisions. LEGSEC reaches 34 MB/s, representing a 70% improvement over baseline. This increase results from parallel decryption execution and reduced re-encryption overhead. Since throughput measures the volume of data processed per unit time, the observed improvement demonstrates that integrating cryptographic transformation with edge collaboration substantially enhances practical performance for large-scale datasets.

Finally, migration cost reduction reflects the efficiency of resource reallocation decisions. The baseline system does not incorporate cost optimization, resulting in no measurable reduction. VMigrate+ achieves a 25.6% reduction through latticetheoretic optimization. LEGSEC further improves this to 31.8%. The additional gain arises because LEGSEC incorporates cryptographic workload characteristics into its utility function, thereby preventing unnecessary task reassignment and minimizing redundant computation. This indicates that combining security-aware scheduling with cost-sensitive optimization yields better economic efficiency than migration-aware strategies alone.

Overall, the experimental results confirm that LEGSEC consistently outperforms both baseline and VMigrate+-style approaches across all measured metrics. The

Citation: Somasekhar Gubbala, Praveen Kumar Kumbum. (2023) Lattice-Guided Secure Edge Collaboration: An Efficient IBE-to-ABE Re-Encryption Framework over Recycled Smartphone Edge Clusters in Temporal Cloud Environments. International Journal of Artificial intelligence and Machine Learning, 1(4), 1-8. https://doi.org/10.55124/jaim.v1i4.302

improvements in re-encryption latency, SLA compliance, CPU utilization, throughput, and migration cost collectively demonstrate the benefits of integrating proxy re-encryption, lattice-guided scheduling, and distributed edge collaboration within a unified framework. LEGSEC improves CPU utilization by adaptive lattice scheduling and reduces SLA violations via constraint-aware allocation.

Comparison with Existing Works

Table 2 presents a structured comparison between prior research efforts and the proposed LEGSEC framework across three critical dimensions: secure sharing capability, edge resource reuse, and lattice-based optimization modeling. These dimensions were selected because they collectively determine whether a system can provide secure, scalable, and SLA-aware distributed computation in dynamic cloud-edge environments.

The IBE-to-ABE conversion work focuses exclusively on cryptographic secure data sharing. It provides an efficient proxy re-encryption mechanism that transforms identity-based ciphertext into attribute-based ciphertext without exposing plaintext to the cloud. While this ensures fine-grained access control and confidentiality, the model assumes a traditional cloud setting and does not incorporate distributed edge reuse or resource-aware scheduling. Additionally, it lacks any formal optimization framework such as lattice modeling for state convergence. Therefore, its strength lies in security, but it does not address orchestration or economic optimization challenges.

MobiTest, in contrast, emphasizes edge reuse by leveraging discarded smartphones to form a collaborative computing cluster. It demonstrates effective scheduling using particle swarm optimization and improves CPU utilization across heterogeneous devices. However, MobiTest does not integrate cryptographic secure sharing mechanisms, nor does it consider fine-grained access control policies. Furthermore, it does not adopt a formal lattice-theoretic model for convergence or SLA guarantees. Its primary contribution is infrastructure-level resource utilization rather than secure data management or theoretical optimization guarantees.

VMigrate+ introduces a lattice-theoretic framework for SLA-compliant virtual machine migration. It models system states within a partially ordered set and guarantees convergence toward Pareto-optimal configurations. This provides strong theoretical backing for cost-aware migration decisions. However, VMigrate+ does not incorporate secure sharing mechanisms such as proxy re-encryption or attribute-based access control. Additionally, while it partially addresses edge dynamics through migration modeling, it is primarily designed for VM-centric cloud infrastructures rather than heterogeneous smartphone clusters with cryptographic workloads.

LEGSEC uniquely integrates all three dimensions. First, it supports secure sharing through efficient IBE-to-ABE proxy re-encryption with verifiable outsourced decryption. Second, it leverages heterogeneous discarded smartphones as an edge cluster, enabling distributed cryptographic processing and resource reuse. Third, it adopts a lattice-theoretic scheduling framework that models system states and ensures SLAaware optimization with provable convergence properties. By combining cryptographic transformation, distributed edge collaboration, and formal optimization theory into a unified architecture, LEGSEC addresses limitations present in individual prior works.

In summary, existing approaches tend to focus on a single axis—either security, edge reuse, or optimization modeling. LEGSEC differentiates itself by integrating these aspects into a cohesive framework. This multidimensional integration enables secure, efficient, and economically optimized computation in dynamic cloud-assisted edge environments, thereby filling a significant gap in current literature.

Unlike prior works, LEGSEC integrates cryptographic conversion, distributed edge collaboration, and lattice-guided SLA compliance.

Conclusion and Future Work

We presented LEGSEC, a unified framework integrating secure proxy re-encryption, distributed smartphone edge reuse, and lattice-theoretic scheduling. Experimental results demonstrate improved efficiency, SLA compliance, and throughput.

Future work includes, Formal provable security under adaptive adversaries, Realworld deployment on large-scale recycled smartphone clusters, and Integration with blockchain-based audit trails.

References

1. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Workshop on the Theory and Application of Cryptographic Techniques, pp. 47–53 (1984). Springer

2. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP'07), pp. 321–334 (2007). IEEE

3. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In: International Conference on the Theory and Applications of Cryptographic Techniques, pp. 127–144 (1998). Springer

4. Meher, K., Sreenivasa Rao, Y.: Efficient data sharing based on ibe to abe conversion for cloud-assisted environments. In: International Conference on Distributed Computing and Intelligent Technology, pp. 103–119 (2026). Springer

5. Tripathi, P., Dhoriya, V., Sasidhar, K., Mankodi, A.: Mobitest: A testbed for harnessing the computing power of discarded smartphones. In: International Conference on Distributed Computing and Intelligent Technology, pp. 120–134 (2026). Springer

6. Kaur, T., Saxena, A.: When agents are powerful: Black hole search in time-varying graphs. In: International Conference on Distributed Computing and Intelligent Technology, pp. 3–18 (2026). Springer

7. Kumar, R., Mazumdar, B., Mandal, S.: On fixed-parameter tractability of weighted 0–1 timed matching problem on temporal graphs. In: International Conference on Distributed Computing and Intelligent Technology, pp. 19–31 (2026). Springer

8. Chand, P.K., Molla, A.R.: Improved linear-time construction of minimal dominating set via mobile agents. In: International Conference on Distributed Computing and Intelligent Technology, pp. 35–51 (2026). Springer

9. Gorain, B., Mandal, S., Sharma, Y.: Collaborative dispersion of silent robots with different wake-up time. In: International Conference on Distributed Computing and Intelligent Technology, pp. 52–67 (2026). Springer

Citation: Somasekhar Gubbala, Praveen Kumar Kumbum. (2023) Lattice-Guided Secure Edge Collaboration: An Efficient IBE-to-ABE Re-Encryption Framework over Recycled Smartphone Edge Clusters in Temporal Cloud Environments. International Journal of Artificial intelligence and Machine Learning, 1(4), 1-8. https://doi.org/10.55124/jaim.v1i4.302

10. Pyati, P., Kaur, N., Jana, S., Bhattacharya, A., Mandal, P.S.: Separation of unconscious robots with obstructed visibility. In: International Conference on Distributed Computing and Intelligent Technology, pp. 68–84 (2026). Springer

11. Mukhopadhyay, N., Tewari, B.P., De, T.: Vmigrate+: A lattice-theoretic and cost-aware framework for sla-compliant vm migration in dynamic cloud infrastructures. In: International Conference on Distributed Computing and Intelligent Technology, pp. 87–102 (2026). Springer

12. Satya Sukumar Makkapati, et al. "Big Data Analytics for COVID-19 Cases using Machine Learning Algorithms." Journal of Optoelectronics Laser 42.3 (2023): 1-11.

13. Bondy, J.A., Murty, U.S.R.: Graph Theory. Springer, ??? (2008)

14. Lamport, L.: Time, clocks, and the ordering of events in a distributed system. In: Concurrency: the Works of Leslie Lamport, pp. 179–196 (2019)

15. Satya Sukumar Makkapati, N. Nagamalleswara Rao, COVID-19 disease detection using attention based Bi-Directional capsule network model Biomedical Signal Processing and Control Volume 96, Part B, October 2024, 106636

16. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT press, ??? (2022)

17. Ahmad, K.A.B., Ahmad, K., Dulhare, U.N.: Functional Encryption. Springer, ??? (2021)

Citation: Somasekhar Gubbala, Praveen Kumar Kumbum. (2023) Lattice-Guided Secure Edge Collaboration: An Efficient IBE-to-ABE Re-Encryption Framework over Recycled Smartphone Edge Clusters in Temporal Cloud Environments. International Journal of Artificial intelligence and Machine Learning, 1(4), 1-8. https://doi.org/10.55124/jaim.v1i4.302