Open Access and Peer Reviewed Journal ISSN 2394-2231 https://www.ijctjournal.org/

RESEARCH ARTICLE OPEN ACCESS

A HYBRID ENCRYPTION ARCHITECTURE WITH DISTRIBUTED KEY MANAGEMENT FOR CLOUD-BASED DATA DIGITIZATION AT HAI PHONG UNIVERSITY

Nguyen Van Quang[1], Hoang Van Lam[1], Tran Bien Thuy[1]

[1]*Faculty of Information Tehcnlogy, Hai Phong University. Haiphong, Vietnam*

Emails: quangnv@dhhp.edu.vn; lamhv@dhhp.edu.vn; thuytb@dhhp.edu.vn

Received: Revised: Accepted: Published:

Abstract - *The authors' earlier study compared the performance of AES, RSA, and a hybrid encryption scheme on digitized data, finding the hybrid model to be the best performance trade-off; however, real-world effectiveness depends heavily on how encryption keys are organized, distributed, and protected across multiple university units. This paper proposes a five-layer architecture — digitization application layer, hybrid encryption layer, Key Management Service (KMS), Role-Based Access Control (RBAC) layer, and cloud storage infrastructure — together with a key lifecycle process (generation, distribution, use/rotation, revocation, audit logging). The team built a simulation model of KMS workload as the number of managed users/records grows from 100 to 10,000, measuring key rotation time, distribution latency, access-control latency, revocation latency, and key-metadata storage size. All figures are simulated data, disclosed transparently to illustrate the scalability of the proposed architecture rather than measurements from a production system. Results show that key-rotation cost and revocation latency scale nearly linearly with system size, while access-control latency grows more slowly thanks to caching, indicating the architecture scales adequately for the university's current and near-term (3–5 year) needs. The paper also provides a conceptual-level threat analysis and mitigation recommendations for insider key-leakage risk.*

Keywords *- encryption key management; cloud security architecture; hybrid encryption; role-based access control; data digitization*

1. Introduction

In prior work, the authors evaluated the simulated performance of AES-256, RSA-2048, and a hybrid AES+RSA scheme, concluding that the hybrid model best balances throughput and key-manageability for Hai Phong University's data-digitization system [1, 2]. In practice, however, the effectiveness of an encryption solution depends not only on the algorithm but heavily on how keys are organized, distributed, and protected throughout their lifecycle — particularly when multiple units, such as the Academic Affairs Office, Student Affairs Office, and academic faculties, share access to the data [3-5].

This paper builds on that result by proposing a concrete system architecture for key management and access control, and evaluates its scalability through a workload simulation model at varying user scales. This represents a shift from algorithm-level evaluation to system-level design, in line with the project's stated goal of proposing a combined solution.

2. Related work

Recent international research on cloud security increasingly emphasizes the central role of a Key Management Service (KMS) within hybrid encryption architectures, along with distributed key-management techniques combined with quantum key distribution or hyperelliptic-curve cryptography for enhanced security [6-8]. In Vietnam, university- and institute-level data-security studies have largely focused on algorithm selection, while operational aspects — key lifecycle management, role-based access control, and audit logging — remain less quantitatively analyzed [9-10]. This paper aims to address that gap within the specific context of Hai Phong University.

3. Proposed Architecture

The proposed architecture comprises five layers, illustrated in Figure 1, organized around the principle of separation of concerns between business logic, data encryption, key management, and access control.

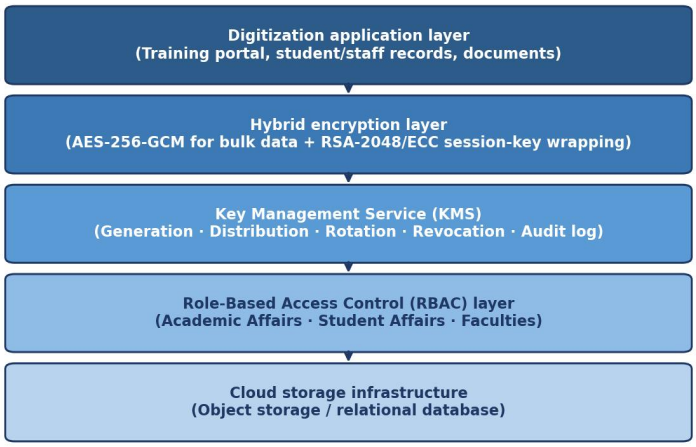

Figure 1. Proposed layered architecture for the cloud data-digitization system

- Digitization application layer: business operations for student/staff records and training documents, where data read/write requests originate.
- Hybrid encryption layer: applies AES-256-GCM for bulk data encryption and RSA-2048 (or ECC) for session-key wrapping, consistent with the validated results of the prior study.
- Key Management Service (KMS): the central component responsible for key generation, distribution, periodic rotation, on-demand revocation, and audit logging of all key-related operations.
- Role-Based Access Control (RBAC) layer: governs key and data access by organizational role (Academic Affairs, Student Affairs, Faculties), enforcing the principle of least privilege.
- Cloud storage infrastructure: stores encrypted data as object storage or a relational database.

The key-management lifecycle is organized into five cyclical steps, shown in Figure 2: key generation, distribution to authorized services/units, use in operations with periodic rotation to bound each key's validity period, revocation when risk is detected or access rights are removed, and audit logging for traceability.

Figure 2. Key-management lifecycle within the proposed architecture

4. Simulation Model and Scalability Evaluation

4.1. Simulation Results

To assess the scalability of the proposed architecture, the team built a workload simulation of the KMS as the number of managed users/records increases from 100 to 10,000 — spanning the university's current scale and its projected growth over the next 3–5 years. Baseline parameters were derived from typical characteristics of commercial KMS offerings and related technical literature (see References), with random noise added to reflect realistic operational variation.

As in the previous paper, it must be stated clearly: all figures in Section 4 are simulated data used to illustrate the architecture's scalability model, not measurements from a live system.

Table 1. Simulated key-management workload versus system scale

Users / Managed Keys	Key Gen. (ms/key)	Distribution (ms)	Access Control (ms)	Key Rotation (s/cycle)	Revocation (ms)	Metadata Storage (KB)
100	3.91	40.97	1.9	2.2	34.13	70.4
500	3.4	56.4	2.25	9.76	46.15	331.6
1000	3.55	56.32	2.66	19.27	51.66	696.1
2500	3.44	64.54	4.07	50.04	69.18	2000.2
5000	3.28	71.35	6.29	99.27	108.78	3615.9
10000	2.71	72.75	10.89	200.35	179.65	7052.1

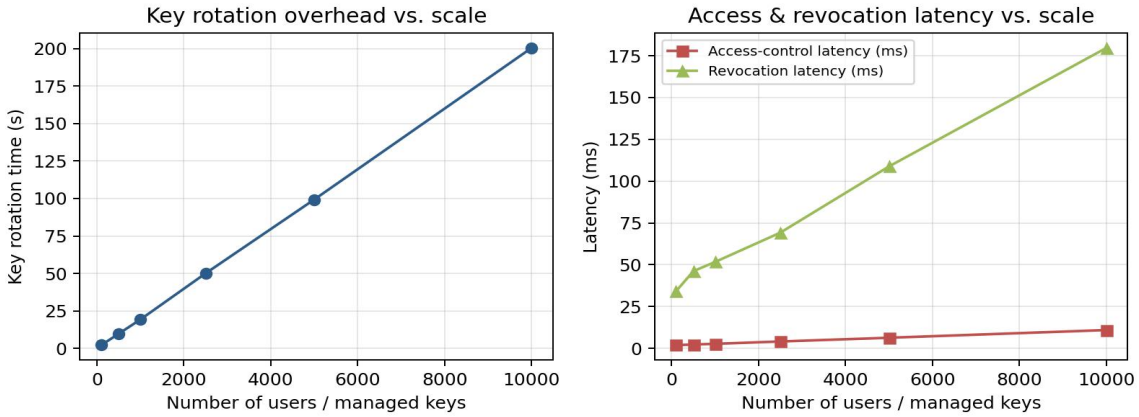

Figure 3. Key-rotation cost and access/revocation latency versus scale

Results show that single-key generation time (3–4 ms) remains essentially constant with scale, consistent with per-key independent operations. In contrast, full-cycle key-rotation time grows nearly linearly, from 2.2 seconds at 100 users to about 200 seconds at 10,000 users, as the number of keys processed concurrently increases. Access-control latency grows far more slowly (from 1.9 ms to 10.9 ms) due to permission caching at the RBAC layer, while revocation latency grows faster (from 34 ms to 180 ms) because revocation state must propagate across KMS cluster nodes. Key-metadata storage (including audit logs) scales linearly with the number of managed keys, reaching about 7 MB at 10,000 users — a negligible storage cost relative to current cloud infrastructure.

4.2. Scalability Discussion

At Hai Phong University's current scale (an estimated few thousand student and staff records), all simulated metrics fall within acceptable bounds for real-time operation: access-control latency under 10 ms and revocation latency under 200 ms. However, the near-linear growth in key-rotation cost suggests that a key-sharding strategy by organizational unit (Office/Faculty) should be considered once scale exceeds roughly 10,000 records, to avoid service disruption from mass key rotation.

5. Conceptual threat – Model Analusis

At a conceptual level, three main risk categories should be mitigated by the proposed architecture: (i) insider key leakage — mitigated through least-privilege enforcement at the RBAC layer and separation of KMS administration from business-data access roles; (ii) key compromise due to insecure storage — mitigated through periodic rotation and storing root keys in a Hardware Security Module (HSM) or a certified KMS offering; and (iii) loss of traceability during an incident — mitigated through an immutable audit-log layer covering all key-related operations. This analysis is conceptual and intended to guide design; formal security proof is outside the scope of this paper.

6. Conclusion and Recommendations

This paper proposed a layered architecture combining hybrid encryption with a centralized key-management service for the cloud data-digitization system at Hai Phong University, and evaluated its scalability via a workload simulation model. Results indicate the architecture comfortably supports the university's current scale and can extend to roughly 10,000 managed records without significant access-control latency. The team recommends: (1) piloting the KMS on live infrastructure to validate and calibrate the simulation model; (2) adopting a per-unit key-sharding strategy once scale exceeds 10,000 records; (3) integrating a Hardware Security Module or certified KMS offering for root-key protection; and (4) formalizing an

RBAC policy tied to the university's data-governance procedures.

References

[1] Le Dac Nhuong (2018), Data Security Textbook, Vietnam National University, Hanoi Press.

[2] Ngo Ba Hung, Thai Minh Tuan, Trieu Thanh Ngoan (2021), Cloud Computing Textbook, Can Tho University Publishing House, ISBN 978-604-965-542-5.

[3] Van, T. H., Vinh, N. P., Ngan, V. T. T., Phuong, L. H., & Le, D. N. (2026). A Course-Specific Agentic RAG Chatbot for IT Student Support: Architecture, Local Deployment, and Preliminary Evaluation at Hai Phong University. *Next-Generation Computing Systems and Technologies*, *2*(2), 21-34.

[4] Ghonge M.M., Pramanik S., Mangrulkar R., Le D.-N. (2022), Cyber Security and Digital Forensics, Wiley, ISBN 9781119795636.

[5] Le D.-N., Bhatt C., Madhukar M. (2019), Security Designs for the Cloud, IoT, and Social Networking, Wiley, ISBN 9781119592266.

[6] Blessing M. (2024), "Cloud Encryption Strategies and Key Management".

[7] Dhinakaran D. et al. (2024), "Towards a Novel Privacy-Preserving Distributed Multiparty Data Outsourcing Scheme for Cloud Computing with Quantum Key Distribution".

[8] Mukhopadhyay D. et al. (2014), "Securing the Data in Clouds with Hyperelliptic Curve Cryptography".

[9] NIST SP 800-57 Part 1 Rev. 5 (2020), Recommendation for Key Management.

[10] NIST FIPS 197 (2001), Advanced Encryption Standard (AES).