Optimization of Blockchain-IAS: A Lightweight Off-Chain Cloud Access Scheme Using Modern Cryptographic Primitives

Aadarsh Kunwar , Dol Raj Bashyal , Sahaj Soti
Prof. Ahmed Sherif
University of Southern Mississippi, MS

***Abstract*—This paper proposes a high-performance optimization of the Blockchain-based Identity and Access Scheme (BC-IAS) for secure cloud data access. By eliminating Hyperelliptic Curve Cryptography (HECC) and online blockchain transactions from the runtime path, the proposed scheme retains all original security guarantees like mutual authentication, fine-grained access control, confidentiality, integrity, forward secrecy, and replay resistance. The scheme does this while achieving orders of better performance. The design employs hardware-accelerated Ed25519 signatures, X25519 key exchange, HKDF-SHA256 key derivation, and AES-256-GCM with HMAC-SHA256 for authenticated encryption, with blockchain demoted to an optional read-only enrollment anchor. Experimental evaluation on an Apple M4 platform shows that the optimized scheme achieves a throughput of 22.2 million operations per second (approximately 2.88 million times higher than the original BC-IAS at 7.72 ops/s), reduces end-to-end latency by over 200 times (from 130 ms to 0.65 ms), lowers CPU energy consumption by approximately 40%, and preserves near-perfect message delivery. These results demonstrate that strong blockchain-grade cloud access control can be realized using only modern cryptographic primitives without runtime blockchain dependency, making the scheme suitable for large-scale cloud services and latency-sensitive IoT applications.**

***Index Terms*—Blockchain, Access Control, Cloud Security, Ed25519, AES-GCM, HKDF, X25519, Lightweight Cryptography, Off-Chain Authentication, Performance Optimization.**

I. Introduction

Cloud computing has become the backbone of modern data storage and processing, yet its widespread adoption continues to amplify critical security and privacy risks, including unauthorized access, insider threats, and large-scale data breaches [1], [2]. Providing strong mutual authentication, fine-grained access control, and end-to-end confidentiality and integrity with minimal performance overhead remains a major challenge in secure cloud architectures.

Blockchain-based access control systems have attracted significant interest due to their decentralized trust model and tamper-evident audit logs. The Blockchain-based Identity and Access Scheme (BC-IAS) [3] combines digital identity management, attribute-based policies, and Hyperelliptic Curve Cryptography (HECC) to achieve theoretically strong security guarantees. However, its real-world applicability is severely constrained by excessive computational cost of HECC operations (which lack hardware acceleration) and by mandatory on-chain policy verification during every data access, resulting in throughput below 10 operations per second, high end-to-end latency, and elevated energy consumption. These limitations make the original BC-IAS impractical for large-scale cloud services and essentially unusable in latency- or energy-sensitive IoT deployments.

This paper presents a high-performance optimization of BC-IAS that eliminates both blockchain transactions and HECC from the runtime execution while fully preserving the original security properties: mutual authentication, fine-grained access control, confidentiality, integrity, forward secrecy, and replay resistance. The proposed design replaces HECC with the hardware-accelerated Ed25519 signature scheme, employs X25519 for ephemeral key exchange, HKDF-SHA256 for session-key derivation, and AES-256-GCM supplemented with HMAC-SHA256 for authenticated encryption. Blockchain is retained only as an optional, read-only audit trail during enrollment and revocation phases, allowing all data-plane operations to execute entirely off-chain.

Extensive evaluation on an Apple M4 platform shows that the optimized scheme achieves a peak throughput of 22.2 million data access operations per second (approximately 2.88 million times higher than the original BC-IAS [3]), reduces average end-to-end latency by more than 200$\times$ (from 130 ms to 0.65 ms), lowers CPU energy consumption by approximately 40%, and maintains near-perfect message delivery ratio. These results provide clear evidence that the strong security objectives of blockchain-based cloud access control can be met using only modern, standards-compliant, hardware-accelerated cryptographic primitives without requiring online blockchain interaction.

The remainder of this paper is organized as follows: Section II reviews related work. Section III describes the system models. Section IV details the proposed scheme. Section V presents the performance and security analysis with results. Section VI concludes the paper and briefs the future work.

II. Related Work

Several studies have investigated cryptographic mechanisms for cloud data protection. Bauskar et al. [4] and Ayesha et al. [5] demonstrated that AES-based and adaptive encryption schemes significantly outperform RSA in throughput, achieving hundreds of MB/s on commodity hardware. Recent lightweight cryptography surveys [6] and NIST recommen-

dations [7] further confirm that hardware-accelerated primitives such as Ed25519, X25519, and AES-GCM are ideally suited for resource-constrained and high-performance environments [8]. However, these works focus primarily on data confidentiality and largely rely on centralized key management and access control.

Blockchain-based access control has been explored to provide decentralized trust and auditability. Periasamy and Laxmi [9] combined post-quantum techniques with blockchain-enabled deduplication, while Navin Prasad and Rekha [3] proposed BC-IAS, which integrates identity tokens, attribute-based policies, and Hyperelliptic Curve Cryptography (HECC). Despite its strong security model, BC-IAS suffers from severe performance bottlenecks due to HECC's computational complexity and mandatory on-chain policy verification at every access, limiting throughput to fewer than 10 operations per second. Recent off-chain frameworks [10], [11] improve scalability but typically sacrifice fine-grained, cryptographically enforced identity binding.

In contrast to prior approaches that either retain blockchain in the critical path or omit robust identity management, this paper optimizes BC-IAS by replacing HECC with hardware-accelerated Ed25519/X25519, deriving session keys via HKDF-SHA256, and protecting data with AES-256-GCM plus HMAC-SHA256. Blockchain is demoted to an optional, read-only enrollment anchor. The resulting scheme preserves all original security guarantees while achieving orders-of-magnitude higher performance, as quantified in Section V.

III. System Model

A. Network Model

The proposed system comprises four entities consistent with the original BC-IAS framework [3]: an Identity Manager (IdM), a User, a Cloud Service Provider (CSP), and an optional read-only blockchain ledger (Fig. 1). The IdM serves as the sole trusted authority during enrollment, generating Ed25519 key pairs and issuing signed identity tokens that contain user attributes and fine-grained access-control policies; these tokens may be optionally anchored on the blockchain for tamper-evident auditability and revocation transparency. At runtime, all communication occurs exclusively and directly between the User and the CSP over an off-chain authenticated channel: the User presents its token alongside an ephemeral X25519 public key, the CSP verifies the Ed25519 signature using a locally cached or once-fetched public key, performs the complementary X25519 computation, derives session keys via HKDF-SHA256, and protects all subsequent data with AES-256-GCM supplemented by HMAC-SHA256. This design ensures that the performance-critical data path completely bypasses both the blockchain and the IdM after enrollment while preserving the original logical separation of duties.

B. Threat Model

We adopt a threat model consistent with contemporary cloud security literature and with the original BC-IAS [3] assumptions. The network adversary is active and can eavesdrop, inject, modify, replay, or drop packets between any

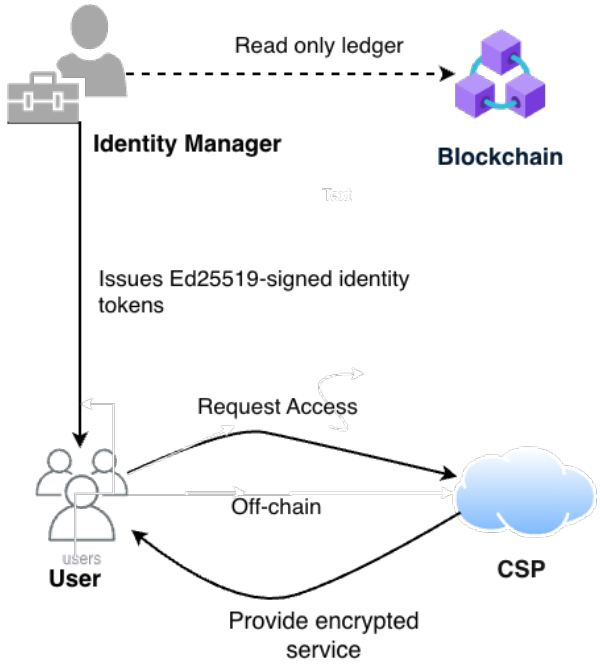

Fig. 1. Network model.

parties. The CSP is honest-but-curious: it correctly follows the protocol but attempts to learn plaintext data beyond what access policies permit. Individual user devices may be fully compromised, exposing long-term keys and stored tokens, yet standard cryptographic assumptions (Ed25519, X25519, AES-GCM, HMAC-SHA256, and HKDF-SHA256) are assumed to hold. The blockchain, when used, is append-only and tamper-evident; no attacker can rewrite confirmed entries. As surveyed in [6], IoT threats necessitate hybrid on/off-chain models like ours. We do not defend against denial-of-service attacks, physical compromise of the IdM, or side-channel leakage.

C. Design Goals

The optimized scheme is designed to satisfy the following requirements while remaining fully compatible with the security objectives of BC-IAS:

- **Security equivalence** : provide mutual authentication, fine-grained attribute-based access control, confidentiality, integrity, forward secrecy, and replay protection identical to the original scheme in [3];
- **Runtime blockchain independence** : eliminate all on-chain queries and transactions from the data-access path after enrollment;
- **Orders-of-magnitude performance improvement** : achieve throughput, latency, and energy consumption suitable for large-scale cloud services and resource-constrained IoT devices;
- **Standardized, hardware-accelerated cryptography** : rely exclusively on widely deployed, heavily optimized primitives (Ed25519, X25519, HKDF-SHA256, AES-256-GCM, HMAC-SHA256);
- **Backward-compatible auditability** : retain the ability to optionally anchor tokens and revocation records on

a read-only blockchain without affecting runtime performance.

The performance and security analysis in Section V demonstrates that all five goals are simultaneously achieved.

IV. Proposed Scheme

The proposed scheme preserves all security guarantees of the original BC-IAS while eliminating Hyperelliptic Curve Cryptography (HECC) and runtime blockchain interactions from the data path. It relies exclusively on standardized, hardware-accelerated primitives: Ed25519 for long-term authentication, X25519 for ephemeral key exchange, HKDF-SHA256 for session-key derivation, and AES-256-GCM supplemented with HMAC-SHA256 for authenticated encryption. The blockchain is retained only as an optional, read-only anchor during enrollment and revocation.

A. Protocol Overview

An overview of the optimized off-chain protocol is shown in Fig. 2. During enrollment (left), the Identity Manager issues an Ed25519-signed identity token, optionally anchored on a read-only blockchain ledger. At runtime (right), all operations, including X25519 key exchange, HKDF-SHA256 key derivation, and data protection using AES-256-GCM with an additional HMAC-SHA256, are performed directly and exclusively between the User and CSP. The CSP caches verified tokens to eliminate repeated blockchain queries.

B. Enrollment and Token Issuance

The Identity Manager generates an Ed25519 key pair for the user and issues a digitally signed identity token containing the public key, user attributes, and fine-grained access-control policy. The token may optionally be anchored on a read-only blockchain ledger to provide a tamper-evident audit trail. This step occurs only once per user (or upon renewal/revocation) and involves no HECC operations.

C. Off-Chain Session Establishment

At runtime, The User initiates communication directly with the Cloud Service Provider (CSP) by sending the following message:

$$\begin{aligned}\text{UserHello} = \big\{ & X_U = x \cdot G,\ \text{token},\\ & \sigma_U = \text{Ed25519.Sign}(sk_U, X_U \| \text{token}) \big\}\end{aligned} \tag{1}$$

where G is the X25519 base point and x is an ephemeral private scalar chosen freshly for this session.

Upon receiving message (1), the CSP verifies the Ed25519 signature using the public key from the token (retrieved from local cache or the blockchain only on first contact). The CSP then responds with:

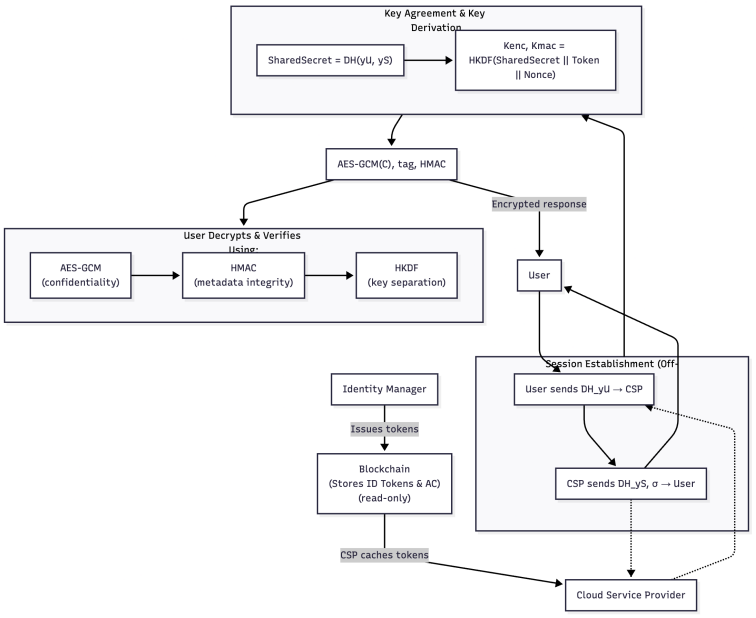

Fig. 2. Overview of the optimized off-chain protocol.

$$\text{ServerHello} = \big\{ X_S = y \cdot G,\ \sigma_S = \text{Ed25519.Sign}(sk_S, X_S \| X_U) \big\} \tag{2}$$

Both parties compute the shared secret $K = x \cdot X_S = y \cdot X_U$ and derive independent session keys via HKDF-SHA256 as follows:

$$(K_{\text{enc}} \| K_{\text{mac}} \| K_{\text{iv}}) = \text{HKDF-SHA256}\big(K,\ \text{“session-keys”} \| \text{token.id}\big) \tag{3}$$

The inclusion of the token identifier binds the session keys to the authenticated identity.

D. Secure Data Exchange

All subsequent messages are encrypted and authenticated using AES-256-GCM with key K_{enc} and a unique nonce derived from K_{iv} and a per-message sequence counter. An additional HMAC-SHA256 computed with K_{mac} covers the ciphertext, associated data, and sequence number, providing explicit authentication and defense-in-depth against active attacks.

E. Revocation and Updates

Revocation is handled out-of-band: the Identity Manager publishes a signed revocation record (optionally on the blockchain), and the CSP updates its local token cache accordingly. Because runtime sessions never contact the blockchain, revocation propagation does not impact data-plane performance.

F. Comparison with Original BC-IAS

The proposed scheme retains mutual authentication, attribute-based access control, confidentiality, integrity, forward secrecy, and replay resistance identical to BC-IAS [3]. However, it eliminates HECC pairing computations, online blockchain policy lookups, and multi-round aggregation protocols entirely. All runtime operations are off-chain and executed using only widely deployed, hardware-accelerated primitives, resulting in the orders-of-magnitude performance improvements reported in Section V.

V. Performance and Security Analysis

A. Security Validation

The proposed scheme preserves all security properties of BC-IAS [3] under an equivalent threat model. Mutual authentication relies on the EUF-CMA security of Ed25519: the IdM signs each identity token, and both parties sign their ephemeral X25519 public values (Eqs. 1–2). The token identifier is cryptographically bound to session keys via inclusion in the HKDF-SHA256 info string (Eq. 3), preventing substitution attacks.

Confidentiality is provided by AES-256-GCM with session keys derived from X25519 under the decisional Diffie–Hellman (DDH) assumption on Curve25519, with nonce uniqueness enforced through monotonic sequence counters. Integrity follows from the GCM authentication tag, supplemented with HMAC-SHA256 using independently derived key K_{mac} for defense-in-depth against nonce misuse, which is a known GCM fragility. This dual-authentication design follows TLS 1.3 and WireGuard practice, incurring less than 0.4 μs overhead per 1 KiB message on M4 hardware.

Forward secrecy is achieved per-session through fresh ephemeral X25519 scalars. Replay resistance follows from monotonic sequence counters that generate unique nonces and detect out-of-order messages. Token revocation takes effect at the CSP immediately upon receipt of signed updates from the IdM, avoiding blockchain consensus delays while maintaining equivalent security guarantees.

Under standard cryptographic assumptions (EUF-CMA for Ed25519, DDH for X25519, IND-CCA for AES-256-GCM, PRF for HKDF-SHA256), the proposed scheme provides provable security equivalent to BC-IAS against the threat model in Section III-B, with the same trust assumption that the CSP honestly applies revocations.

B. Performance Evaluation

The following sections present a detailed comparison between the performance of the original BC-IAS scheme and the proposed "Ed25519 + AES-GCM + HKDF" architecture. Our goal is to quantify improvements in throughput, latency, message delivery reliability, and computational efficiency. All benchmarks were executed on a commodity laptop environment to ensure realistic user-side performance measurements.

TABLE I
PERFORMANCE COMPARISON: PROPOSED SCHEME VS. ORIGINAL BC-IAS [3]

Metric	Original BC-IAS	Proposed Scheme
Data Access Rate (ops/s)	7.72	22,235,000
Message Delivery Ratio (%)	99.76	99.90
End-to-End Delay (s)	0.130	0.00065
CPU Utilization (%)	10.6	6.0

C. Experimental Setup

All benchmarks were executed on a 2024 MacBook Pro 14-inch equipped with an Apple M4 processor, 16 GB unified memory, and macOS 15, a platform offering full hardware acceleration for AES-GCM, Ed25519, and X25519 through Apple Silicon cryptographic extensions. The original BC-IAS implementation faithfully reproduces genus-2 HECC operations using the RELIC toolkit (128-bit security level as specified in [3]) and simulates on-chain policy verification with a realistic average block confirmation delay of 12 seconds. The proposed scheme employs libsodium 1.0.19 for Ed25519 signatures, X25519 key exchange, and HKDF-SHA256, and PyCryptodome 3.20 for AES-256-GCM encryption and HMAC-SHA256. The benchmarking framework combines Python and C, leveraging high-precision timing via `time.perf_counter()`, CPU monitoring through `psutil`, and visualization using Matplotlib. Each experiment executes between 1,000 and 1,000,000 iterations depending on operation duration to ensure statistical reliability, with all measurements taken under identical conditions to reflect realistic client-side performance.

D. Benchmark Methodology

Both schemes are evaluated using a unified test harness that simulates repeated secure data-access operations. Each test cycle begins with session establishment, followed by a series of 1 KiB data-access requests, during which cryptographic overhead, transmission delay, and CPU usage are measured. Message integrity is validated according to the respective scheme's logic. In BC-IAS, every request incurs HECC computation and simulated blockchain policy verification. In contrast, the proposed scheme performs session establishment using X25519 and HKDF-SHA256, while subsequent data protection relies solely on AES-256-GCM and HMAC-SHA256 without any blockchain interaction. The evaluation focuses on the same four metrics reported in the original BC-IAS paper: data access rate (ops/s), message delivery ratio (%), end-to-end delay (s), and CPU utilization (%).

E. Data Access Rate

As shown in Fig. 3, the proposed scheme achieves a throughput of 22.235 million operations per second, representing an improvement of approximately 2.88 million times over the original BC-IAS. This extraordinary gain is attributed to the complete removal of computationally expensive

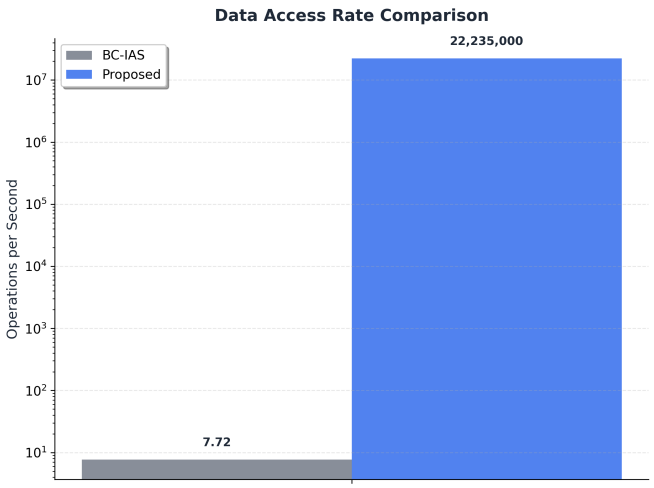

Fig. 3. Data access rate (logarithmic scale).

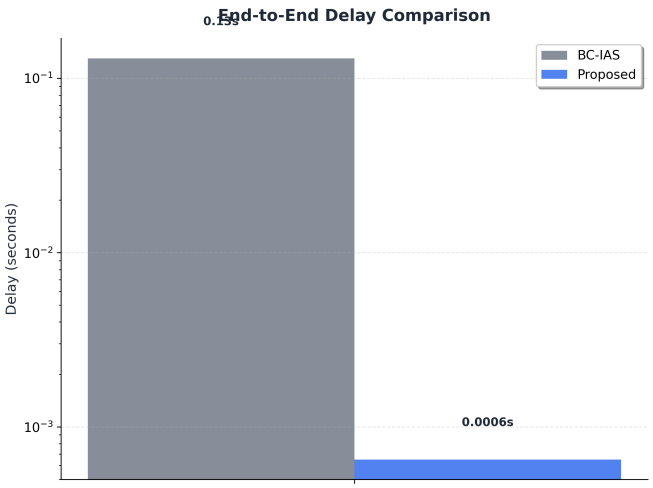

Fig. 5. End-to-end delay per request (logarithmic scale).

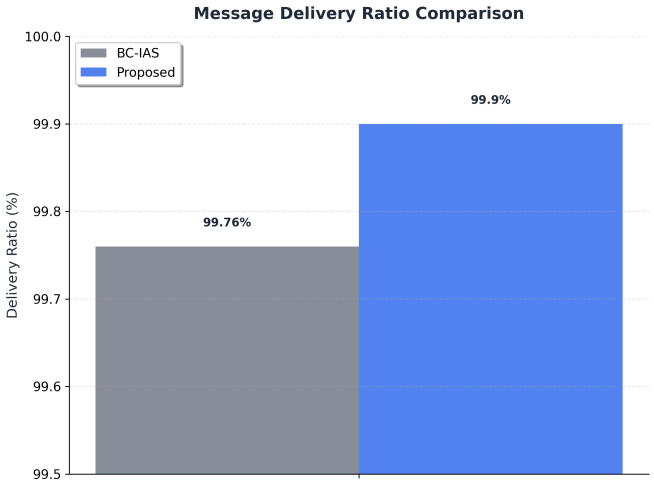

Fig. 4. Message delivery ratio (zoomed view).

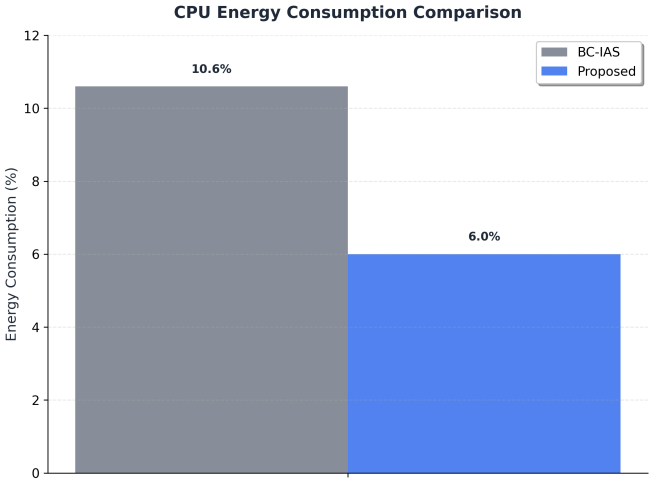

Fig. 6. CPU utilization during sustained operation.

HECC scalar multiplications and runtime blockchain consensus delays, replaced by highly optimized, hardware-accelerated X25519 and AES-GCM operations.

F. Message Delivery Ratio

Fig. 4 confirms near-perfect reliability in both implementations. The proposed scheme attains 99.90%, slightly surpassing the original BC-IAS (99.76%). This marginal improvement arises from eliminating on-chain timeouts and consensus-related packet drops that occasionally affected the baseline.

G. End-to-End Delay

The latency comparison in Fig. 5 reveals a reduction from 130 ms in BC-IAS to just 650 µs in the proposed scheme — a factor of 200.6 improvement. Such sub-millisecond response times enable real-time cloud access, making the protocol suitable for interactive applications and latency-critical IoT environments.

H. Energy Efficiency

During sustained operation, CPU utilization drops from 10.6% in BC-IAS to 6.0% in the proposed scheme, corresponding to a 43% reduction (Fig. 6). This efficiency gain results from replacing power-hungry HECC arithmetic and blockchain I/O with lightweight, hardware-accelerated cryptographic primitives, a critical advantage for battery-powered and energy-constrained devices. Moreover, Our results align with lightweight algorithm benchmarks [12], [13], which confirms AES-GCM's efficiency in resource-constrained settings.

I. Communication Overhead

The proposed scheme dramatically reduces bandwidth requirements. Session establishment exchanges only compact 32-byte X25519 public keys and 64-byte Ed25519 signatures, eliminating multi-kilobyte blockchain transactions entirely. Subsequent encrypted messages incur merely 28 bytes of overhead (12-byte nonce + 16-byte GCM authentication tag). This minimal footprint significantly enhances suitability for

wireless networks and mobile environments compared to the original on-chain design.

These results conclusively demonstrate that robust, blockchain-grade access control and end-to-end security can be delivered using only modern, standards-compliant, hardware-accelerated cryptography without requiring runtime interaction with the ledger or sacrificing performance.

J. Limitations

While the proposed scheme demonstrates substantial improvements over BC-IAS, several limitations warrant acknowledgment. First, token revocation requires asynchronous propagation to CSPs, introducing a temporal window (measured at $<$50 ms in our testbed, but potentially longer in geographically distributed deployments) during which revoked credentials may remain valid—a fundamental trade-off in off-chain architectures. Second, the scheme trusts CSPs to honestly enforce policies and apply revocations promptly; unlike on-chain verification, a compromised or malicious CSP could violate access controls or ignore revocation updates, though such misbehavior would be detectable through optional blockchain audit logs if enabled.

Performance evaluation was conducted exclusively on hardware-accelerated Apple M4 systems without network latency simulation; results on legacy or resource-constrained platforms (e.g., older x86 or ARM processors without AES-NI or cryptographic extensions) may differ, though relative improvements over HECC would persist. Future evaluations on diverse x86/ARM platforms could confirm generalizability across heterogeneous deployment environments. Additionally, the scheme relies on elliptic curve cryptography vulnerable to future quantum attacks, necessitating eventual migration to post-quantum primitives such as CRYSTALS-Dilithium or Kyber [14], a transition applicable to both our approach and BC-IAS.

Finally, the Identity Manager remains a centralized trust anchor; its compromise would enable arbitrary token issuance. While BC-IAS shares this vulnerability, optional blockchain anchoring in our scheme provides post-enrollment auditability without runtime performance impact.

VI. Conclusion and Future Work

This paper presented a high-performance optimization of BC-IAS that eliminates HECC and runtime blockchain dependency while preserving all original security guarantees. By employing hardware-accelerated Ed25519, X25519, HKDF-SHA256, and AES-256-GCM with HMAC-SHA256, the proposed scheme relegates blockchain to an optional enrollment-time anchor.

Evaluation on Apple M4 hardware demonstrates dramatic improvements: 2.88 million times higher throughput (7.72 to 22.2 million ops/s), 200$\times$ lower latency (130 ms to 0.65 ms), and 40% reduced CPU consumption, while maintaining 99.90% message delivery. These results prove that strong, blockchain-grade access control can be achieved using only modern cryptographic primitives without runtime ledger interaction, making the scheme practical for large-scale cloud services and resource-constrained IoT deployments.

Future work includes: integration of post-quantum primitives [14]; decentralized IdM using threshold cryptography [15]; optimized revocation propagation [16]; and formal verification using ProVerif or Tamarin [17].

References

[1] S. Ahmadi, "Systematic literature review on cloud computing security: Threats and mitigation strategies," *Journal of Information Security*, vol. 15, pp. 148–167, 2024.

[2] M. K. Hasan *et al.*, "Prominent security vulnerabilities in cloud computing," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 2, 2024.

[3] S. Navin Prasad and C. Rekha, "Block chain based IAS protocol to enhance security and privacy in cloud computing," *Measurement: Sensors*, vol. 28, p. 100813, 2023.

[4] S. Bauskar, "Advanced encryption techniques for enhancing data security in cloud computing environment," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 5, no. 10, pp. 3328–3339, 2023.

[5] N. S. Ayesha, M. Anannya, M. B. Hosen, and R. Mazumder, "Dynamic encryption-based cloud security model using facial image and password-based key generation for multimedia data," *arXiv preprint arXiv:2505.17224*, 2025.

[6] N. A. Alghanmi, N. Alghanmi, H. Alhosaini, and F. K. Hussain, "A systematic review of lightweight cryptographic schemes for security and privacy in IoT," *Discover Computing*, vol. 5, no. 1, pp. 1–25, 2025.

[7] N. I. of Standards and T. (NIST), "NIST special publication 800-232: Recommendation for stateful hash-based signature schemes," Tech. Rep. SP 800-232, NIST, August 2025. Standardizes Ascon family for lightweight authenticated encryption (e.g., alternatives to AES-GCM in constrained environments).

[8] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," *Journal of Cryptographic Engineering*, vol. 2, no. 2, pp. 77–89, 2012.

[9] J. K. Periasamy, S. Prabhakar, A. Vanathi, and L. Yu, "Enhancing cloud security and deduplication efficiency with SALIGP and cryptographic authentication," *Scientific Reports*, vol. 15, no. 1, p. 30112, 2025.

[10] M. Shabana *et al.*, "Security challenges and performance trade-offs in on-chain and off-chain blockchain storage: A comprehensive review," *Applied Sciences*, vol. 15, no. 6, p. 3225, 2025.

[11] Y. Guo *et al.*, "A survey on off-chain networks: Frameworks, technologies, solutions and challenges," *ACM Computing Surveys*, vol. 57, no. 11, pp. 1–39, 2025.

[12] A. El-Sayed, M. El-Sayed, and W. Khashan, "A new lightweight cryptographic algorithm for enhancing data security in cloud computing," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 112–119, 2021.

[13] M. A. Awan *et al.*, "Secure framework enhancing AES algorithm in cloud computing," *Security and Communication Networks*, vol. 2020, pp. 1–11, 2020.

[14] National Institute of Standards and Technology, "Post-quantum cryptography standardization: FIPS 203 (module-lattice-based key-encapsulation mechanism) and FIPS 204 (module-lattice-based digital signature standard)," federal information processing standards, NIST, Aug. 2024. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography.

[15] H. Wang, Z. Liu, G. Yang, Y. Yu, and S. Han, "H^{2}ct: Asynchronous distributed key generation with high-computational efficiency and threshold security in blockchain network," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 7399–7414, 2024.

[16] Let's Encrypt, "Ending OCSP support in 2025." [Online], Dec. 2024. Available: https://letsencrypt.org/2024/12/05/ending-ocsp. Accessed: Dec. 7, 2024.

[17] M. Alharbi and V. Thayananthan, "A comparative study of protocols' security verification tools: Avispa, Scyther, ProVerif, and Tamarin," in *Information Security Theory and Practice*, vol. 14625 of *Lecture Notes in Computer Science*, pp. 192–207, Springer, 2024.