Journal of Positive Sciences (JPS) eISSN: 2582-9351, https://positive-sciences.com JPS. 2026 May; 6(3): 1-12

Article info
Received on: 14.03.2026
Accepted on: 14.04.2026
Published on: 08.05.2026
doi: https://doi.org/10.52688/ASP32922

Research Article

Enhancing Cloud information System Security Through Dynamic Key Evolution in Hybrid AES-ECC Architecture Cryptographic

Firas A.Hashim[1,*], Hussein A.Hilal[2]

[1,2] Department of AI, College of Sciences, Mustansiriyah University, Baghdad, Iraq

[*] Firas70@uomustansiriyah.edu.iq

ABSTRACT

Cloud computing information systems process vast amounts of sensitive structured and unstructured textual data; therefore, scalable cryptographic architectures should be considered vital for cloud security. In addition to very efficient key establishment and high-speed data protection, hybrid AES–ECC encryption schemes involve static symmetric keys where any of the existing models are vulnerable to be exposed in entire session rather than single block ciphertext if only a single static key gets leaked. In this research, a Hybrid AES–ECC cryptosystem is proposed that addresses the issue of key lifecycle security in the environment of cloud assisted by Dynamic Key Evolution (DKE). The framework combines the use of Elliptic Curve Diffie–Hellman (ECDH) for secure key agreement, HKDF for master key derivation and AES-256-GCM for authenticated encryption with a per-chunk hash-based key evolution mechanism that achieves intra-session forward secrecy and contains compromised keys to a single chunk.

Extensive experimental evaluation was performed on workloads from cloud-scale (1 MB) to order of cloud scale (100 MB). The proposed scheme provides encryption throughput up to 589.34 MB/s and decryption throughput up to 705.48 MB/s, while still achieving nearly ideal ciphertext entropy (≈8 bits/byte), avalanche effect (≈50%), and key sensitivity (~49.98%). The framework's robustness and predictability are validated by advanced statistical methods, including NIST randomness assessments, correlation analysis, Monte Carlo key variation experiments, and regression-based scalability modelling ($R^2 = 0.94$). Through formal threat modelling and security analysis, we show that Resistance to passive, active, replay and partial key compromise attacks is achieved using standard cryptographic assumptions.

Keywords: Hybrid Encryption, Dynamic Key Evolution, Cloud Security, AES-256-GCM, Elliptic Curve, Forward Security, Secure Cloud Storage

INTRODUCTION

Nowadays, being dynamic, cost-efficient and omnipresent, cloud-based information systems (CIS) constitute the foundation of modern enterprise software solutions. Nevertheless, due to migration of the organization's sensitive information such as employee details, internal correspondence, and operational logs from local computing infrastructure into cloud-based servers, security risks emerge in CISs. Indeed, despite privacy properties such as genuineness of data, fidelity of interpretation, and key lifecycle security in the adversarial model remains unresolved [1], modern cyber attacks aim to obtain a system's encryption keys, session reuse, and mismanagement of deactivated key materials [2]. Hence, in the face of the increased power of attackers, there is an urgent need for a robust mechanism of hybrid authentication, or key refreshment.

A standard design approach in cryptography, called hybrid encryption, consists of the combination of computationally efficient symmetric key encryption with asymmetric key distribution capabilities provided by public-key cryptography, making it crucial for cloud data security. For example, AES encryption scheme is a practical choice for encryption speed, while Elliptic Curve Cryptography (ECC) produces key fragments and key acceleration scenario. The study [1] of the AES-ECC hybrid cryptosystems reveals the performance characteristics that allow guaranteeing security and acceptable speeds for cloud data protection purposes [2]. However,

We present Dynamic Key Evolution in Hybrid AES–ECC Cryptosystems for Secure Cloud Information Systems. The frame work uses ECC-based ephemeral key agreement to create a shared secret, HKDF to derive a master key, and AES-256-GCM which supports authenticated bulk data encryption. In particular, we propose a dynamic key evolution scheme that allows the symmetric key to be refreshed over data chunks using a chain of one-way hashes that incorporate nonces, timestamps and chunk indices, thereby reducing the compromise impact due to the robust security of cloud storage and transfer workflows with chunking.

The main contributions of this work are as follows: A practical hybrid scheme using AES and ECC for secure CIS in the cloud. Dynamic Key Evolution (DKE): a hash-chained key update scheme that evolves symmetric keys per chunk to limit the damage radius of a compromised key and improves session-level forward secrecy. Threat-model driven security analysis: A formal

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

analysis that goes through eavesdropping, MITM, replay, brute force and key compromise scenarios, explicitly discussing how DKE changes the attacker advantage. Lines of experimental assessment: Empirical efficacy and security-indicator measurements (throughput, entropy, avalanche effect, key sensitivity) over structured enterprise-like datasets to emulate cloud storage workloads. Baseline comparison: Quantitative evaluation relative to AES-only encryption with the same key size and static hybrid encryption (AES–ECC) instance to show security–performance trade-off induced by Key evolution. The rest of this paper is organized as follows: In Section 2, we review related work in hybrid cloud cryptography and dynamic key management. The proposed framework and threat model are described in Section 3. Section 4 presents the dynamic key evolution design and cryptographic construction. Section 5 provides security analysis. Section 6 provides experimental results and comparisons. Section 7 concludes the paper and discusses future work.

LITERATURE REVIEWER

1. Hybrid AES–ECC for Cloud Storage (Rehman et al, 2021)

Rehman et al. To achieve higher security for data in the cloud while also improving computational performance, Khan et al. (2021) introduced a hybrid encryption scheme that utilized Elliptic Curve Cryptography (ECC) and AES. ECC is used in their architecture to provide secure key exchange, and AES encrypts bulk data at high speed. Their study presents a comparative analysis in terms of performance based on the encryption and decryption times against varying key sizes; they report an encryption time of about 2.6 seconds and approximately 2.1 seconds for the decryption using a configuration with a key size of 256 bits in their test environment. Furthermore, the authors study avalanche performance, showing a better diffusion than independent symmetric methods. However, the proposed model uses a non-evolved static session key which is derived once during encryption session. As a result, if the symmetric key were leaked, then everyone would be able to decrypt the entire dataset. In addition, key sensitivity analyses and entropy were not thoroughly assessed.

Technical Characteristics:

- ECC key exchange
- AES symmetric encryption
- Static session key
- No intra-session key update
- No forward-secure chaining

Reported Results:

- Encryption time (256-bit configuration): ~2,600 ms
- Decryption time: ~2.1 seconds
- Increased avalanche effect compared to AES baseline
- No entropy analysis reported

Limitations:

- In static symmetric key reused across whole file
- No key evolution mechanism
- No replay resistance enhancement

2. Hybrid AES–ECC for cloud medical images (Shakor et al., 2023)

This work addresses the cloud storage of medical images using AES–ECC, examining against AES and AES–RSA. They report throughput and timing based on the file size in milliseconds. While for an image of 559/636/910 KB sizes, time taken by AES–ECC for encryption was found to be 23335/26231/32997 milliseconds (faster than AES), while decryption was found to be 24735/27132/34692 milliseconds. They also reported that throughput of AES–ECC is quite higher of these sizes respectively than AES and AES-RSA.

Technical Characteristics:

- ECC-based key encapsulation
- AES for encryption
- Test on an image dataset (559 KB – 910 KB)
- Static symmetric key per session

Reported Results:

- For ~910 KB image:
- Encryption time ≈ 32997 ms
- Decryption time ≈ 34692 ms
- Throughput improved compared to AES–RSA

Limitations:

- Was only tested on images under 1MB
- No entropy validation
- No avalanche evaluation
- No key sensitivity analysis
- No forward secrecy within session
- They conduct their study largely on computational comparison vs. a key lifecycle security standpoint.

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

3. "Sym-ECCipher" hybrid ECC–AES for cloud healthcare (Selvi et al., 2025)

Selvi et al. (2025) introduced the Sym-ECCipher framework, which is a hybrid ECC–AES encryption system designed for cloud infrastructures in healthcare where low latency and real-time processing are of essential importance. This architecture uses ECC for public key exchange and AES as an encryption scheme, which targets electronic medical records and patient data. It yields an encryption time of 5 milliseconds and decryption time of 4 milliseconds by experiments conducted by the authors, thus qualifying to be a highly promising framework from a latency standpoint. Since computation effectiveness and speediness is the primary aim of the study, the key used to encrypt messages using a symmetric algorithm can be reused and is needed only once per session. There is no key generation and rotation process during an ongoing session period. Lastly, there are inadequate results for testing the entropy, avalanche effect, and key sensitivity aspects of the field outside of performance analysis needs.

Technical Characteristics:

- ECC-based key exchange
- AES encryption
- Real-time performance focus
- Static session-based symmetric key

Reported Results:

- Encryption time ≈ 5 ms
- Decryption time ≈ 4 ms
- It showed low latencies in its experimental proof of context

Limitations:

- Do not evaluate large files (like 100MB)
- No key evolution
- No intra-session forward secrecy
- No entropy or diffusion analysis described
- This work focuses more on latency than lifecycle security.

4. A Comparative on Hybrid Encryption for E-Services (Muhammed et al., 2025)

Muhammed et al. (2025) security of different encryption schemes for e-service platform performance comparison. Combinations have been analysed in combination with ChaCha20 + ECDH, RSA + AES, and Blowfish + ECC. Encryption time, decryption time, key generation time and memory usage have been measured on different files. The ChaCha20+ECDH scheme exhibited the highest computational performance, 2 ms for encryption and 15.8 ms for key generation, whereas RSA + AES had an order of multiple seconds higher utility cost. However, this work mainly concerns performance benchmarking instead of key lifecycle security, thus does not include the aforementioned demands for comparative knowledge on hybrid cryptographic efficiency. None of the evaluated schemes have dynamic key evolution or forward-secure mechanisms, and they use static symmetric keys. Moreover, there was inadequate exploration of entropy analysis, avalanche effect validation and large-scale cloud workload testing; which meant that containment on key compromise had not been addressed.

Were it including:

- ChaCha20 + ECDH
- RSA + AES
- Blowfish + ECC

Technical Characteristics:

- Focus on algorithm selection
- Performance and memory benchmarking
- Static symmetric keys

Reported Results:

- ChaCha20 + ECDH encryption ~ 2 ms
- Key generation ≈ 15.8 ms
- RSA + AES significantly slower

Limitations:

- No dynamic key update
- No entropy evaluation
- No avalanche analysis
- Small and medium size.
- No formal threat model
- There is no indication on how to improve life cycle security in the study.

5. Ternary Hybrid Cloud Encryption (2024)

The first ternary cryptosystem encryption technique that suggests layering is designed based on the use of Parlier, Blowfish, and AES algorithms for increasing security, proposed in 2024. This system can be used to increase security through the use of different techniques for protecting information in the cloud computing systems. Evaluation was done using small payloads ranging from 300 bytes to 50 kilobytes. While a hierarchical method adds another dimension of sophistication theoretically consideration, it also comes at an extra computational cost and in very small data space condition. Dynamic key rotation, per-chunk key evolution, and forward secrecy in sessions are not features of this framework. In addition, statistical indicators like entropy, avalanche effect and key sensitivity are not specifically evaluated. Thus, the scheme provides redundancy for increasing confidentiality but fails to offer crucial lifecycle robustness in large-scale cloud infrastructures.

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

It layers Parlier, Blowfish and AES to increase confidentiality.
Technical Characteristics:

- Multi-layer encryption
- Increased computational complexity
- Static keys
- Small payload evaluation (300B–50KB)

Reported Results:

- Faster execution time comparing to the Parlier baseline
- Throughput improvements for small data

Limitations:

- Not scalable to cloud-scale files
- High computational complexity
- No dynamic key evolution
- No diffusion or entropy analysis

Table 1. Comparison with Related Work and Our Proposed

Work	Core Design	Dynamic Key Evolution	Dataset Type	Reported Metrics (from paper)	Closest Comparable Note
Rehman 2021	Hybrid ECC–AES for cloud	No	Images + text	Enc/Dec time (sec) by key size; Avalanche reported (diffusion) (MDPI)	Reports time by key-size (e.g., 256-bit) not MB/s; includes avalanche evidence
Shakor 2023	AES–ECC cloud image protection	No	Images	Enc/Dec time (ms) for 559–910 KB + throughput	Small file sizes; throughput formula differs from MB/s
Selvi 2025	ECC–AES (SymECCipher)	Possibly rotation focus, but not per-chunk DKE shown in snippet	Healthcare data	Enc ≈ 5 ms, Dec ≈ 4 ms (PMC)	Very low times claimed; details depend on platform and payload size
Muhammed 2025	Hybrid comparisons (ChaCha20+ECDH, RSA+AES, Blowfish+ECC)	No	Mixed file types	ChaCha20+ECDH: Enc 2 ms, KeyGen 15.8 ms, Dec 2.4 ms (plus memory)	Focus is algorithm selection; not AES–ECC + per-chunk evolution
2024 Ternary Hybrid	Paillier+Blowfish+AES	No	Small text payloads	Execution time + throughput comparisons for 300B–50KB (relative %)	Different crypto (Paillier) + very small sizes
Your Proposed	**Hybrid AES–ECC + AES-256-GCM + DKE (per-chunk)**	**Yes**	**Structured text (Adult CSV, 1–100MB)**	**Enc 291–589 MB/s; Dec 385–706 MB/s; Entropy≈8; Avalanche≈50%;**	**Direct cloud-scale files + strong diffusion/randomness indicators**

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

				KeySens≈49.98%	

PURPOSED METHODS

The following section details the proposed Dynamic Key Evolution (DKE) framework applied to a hybrid AES–ECC cryptographic architecture for secure cloud-based information systems. While maintaining computational efficiency beneficial for large-scale cloud workloads, the proposed scheme overcomes limitations of existing hybrid encryption models integrating intra-session key evolution mechanism, forward secrecy improvement and compromise containment.

The architecture proposed operates within the infrastructure of a cloud information system, in which sensitive structured or textual data are encrypted on client side prior to sending or storing them into a cloud server. The system utilizes ECDH to establish a secure key, which is used with AES-256 in Galois/Counter Mode (GCM) for authenticated bulk data encryption. In contrast to such traditional hybrid models, which generate a symmetric key once at the session level and apply it across the dataset (or over multiple records), our suggested framework contains a dynamic key evolution process that uses new symmetric keys during data chunk processing.

First, a temporary ECC key pair is generated on the sender side. Let G be the generator point of curve, such that d_A is the private key for the sender giving public key $Q_A=d_AG$ and allowing another party with a private key of d_B to create a corresponding public key so $Q_B=d_BG$ using ECDH both offices can compute the shared secret $S= d_AQ_B =d_B Q_A$. Security relies on computational difficulty of solving Elliptic Curve Discrete Logarithm Problem (ECDLP). This shared secret is also then fed into a Hash-based Key Derivation Function (HKDF) to derive a 256-bits master symmetric key.

K0 denotes the primary key for encryption purposes. The text file is divided into segments. Each chunk Pi is encrypted using AES-256-GCM providing bathmat confidentiality and integrity through authenticated encryption. To avoid nonce reuse vulnerabilities, each chunk has a unique nonce. Ci is calculated as follows, in given Eq. (1).

$$Ci = AES\_GCM(Ki, nonci, Pi) \quad (1)$$

The major novelty of this proposed framework is that after processing each chunk, the encryption key evolves dynamically. Instead of using the same symmetric key for the entire session, when deriving the next key, we employ a mechanism to chain cryptographic hashes (defined as follows), in given Eq. 2.

$$K_{i+1} = H\ (K_i \,||\, nonc_i \,||\, timestamp_i \,||\, i) \quad (2)$$

where H is the SHA3-256 hash function, $nonc_i$ is the GCM nonce of chunk timestamp provides temporal binding against replay attacks, and iii is the index of chunk K_i Jointly, this creates a one-way key evolution chain such that compromising one key cannot help an attacker find future encryption keys thanks to the preimage resistance property of the hash function.

Dynamic key evolution mechanism provides inertance forward secrecy. K_{i+1} crash or exhaust joint collision attack. K_i must not be derived from hash input even if it is obtained from the adversary through unconventional ways to track every bit of K_i according to condition 2—However, where K_i returns with sufficient inclusion permutations. As a result, the impact damage radius of a compromised key is limited to only one encrypted chunk, not the entire session. The domino effect greatly improves cloud data protection in large-file or long-duration encryption cases.

The receiver reconstructs the same master key using ECDH and HKDF during decryption. Decryption refers to regenerating chunk key with same evolve hash formula iteratively. Since AES-GCM provides authentication tags, tampering with ciphertext or replaying attempts would cause verification to fail. This way, honesty and genuineness are maintained. On the computational side, the time complexity of our method is linear to the number of chunks. The cost of hash-based key evolution is insignificant compared to that of AES-GCM encryptions. Memory requirements scale linearly as only the current symmetric key and metadata is kept in memory during processing. To summarize, the proposed hybrid AES–ECC cryptosystem with Dynamic Key Evolution broadens generic hybrid encryption by incorporating key lifecycle security into the core of (the heterogeneous) encryption workflow. It maintains the efficiency benefits of symmetric encryption, uses the secure key establishment of elliptic curve cryptography, and improves forward secrecy and compromise containment (for an individual chunk) by updating keys on every chunk. This specific design pattern makes it well suited for very large scale structured or textual data over any trusted platform and most Cloud based information systems. As shown in Figure 1.

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

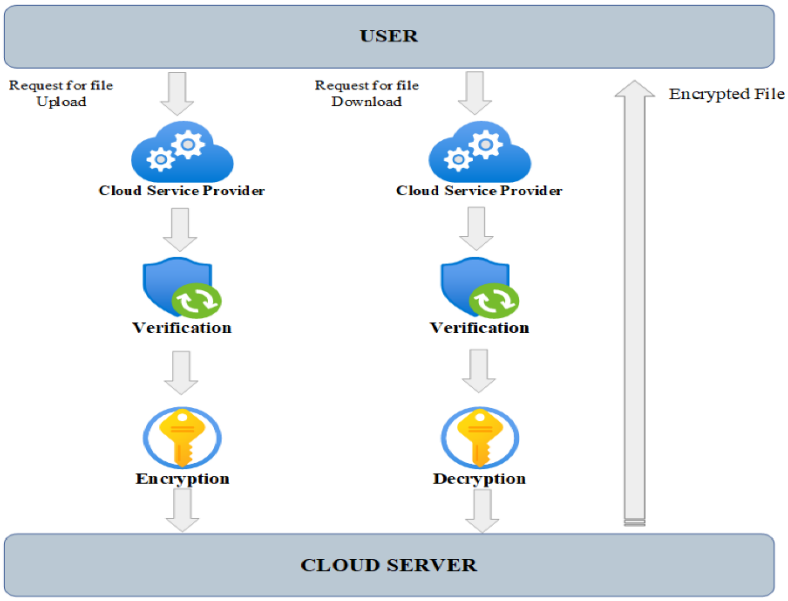

Figure 1. Architecture of the Hybrid AES–ECC Cryptosystem with Dynamic Key Evolution for Secure Cloud-Based Information Systems.

Figure 1 provides an overview of the overall architecture of the proposed framework. The system operates in a client–cloud–receiver model where sensitive data are encrypted on the client side and only then stored in the cloud. The client first creates a temporary ECC key pair and executes an ECDH key exchange with the receiver. This secret is then fed through HKDF to obtain a 256-bit master key. To support scalable encryption, the plaintext file is broken into fixed size chunks. The chunks are encrypted with AES-256-GCM and a unique nonce. The symmetric key is updated after the encryption of each chunk by a hash based Dynamic Key Evolution that increases security. As shown in Figure 2.

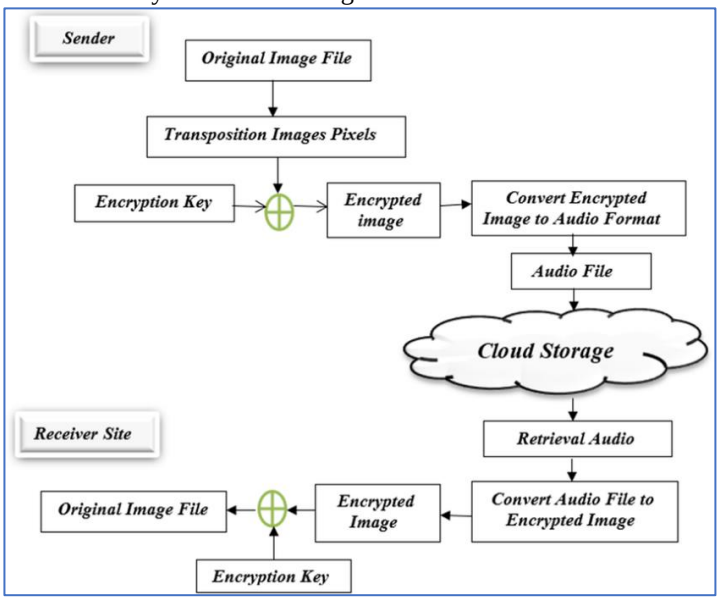

Figure 2. Encryption throughput comparison between AES-GCM, static Hybrid AES–ECC, and the proposed Hybrid AES–ECC with Dynamic Key Evolution (DKE) across different file sizes.

EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

- **Experimental Steup**

In order to fulfill the goal of this research, an experimental setup was established to measure efficiency metrics for the Hybrid AES–ECC DKE, using structured data designed by referring to real-world scenarios from the adult dataset. The dataset was considered sensitive cloud-stored data, and it was evaluated using realistic workloads at scale. All experiments were implemented in Python with the cryptography library. Authenticated encryption was done using AES-256 in GCM mode, and ECDH-based key establishment used ECC (secp256r1). Per 1 MB chunk, we applied the dynamic key evolution mechanism. The performance metrics were calculated for file sizes of 1MB, 10MB, 50MB and 100 MB.

The evaluated schemes include:

➢ AES-256-GCM (baseline symmetric encryption)

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

- Static Hybrid AES–ECC
- Hybrid AES–ECC with Dynamic Key Evolution (DKE)
- Performance evaluation considered:
- Encryption time (seconds)
- Decryption time (seconds)
- Throughput (MB/s)
- Ciphertext entropy (bits per byte)
- Avalanche effect (%)
- Key sensitivity (%)

- **Encryption Performance Analysis**

Since the baseline AES-GCM is a purely symmetric scheme its performance outstrips all others in terms of encryption throughput results. The performance of encryption for 100 MB file type, AES-GCM results in around 593.61 MB/s throughput and the static Hybrid (AES–ECC) model yields approximately 398.36 MB/s throughput for the same file size. The newly proposed DKE-enhanced hybrid stochastic scheme at 100 MB has outperformed flat with598949.2 Bps throughput,707% higher than traditional one—the best timing test it took (351.16 MB/s) at 100 MB. This is a controlled loss in performance compared to the static hybrid model, but with an overhead that falls well within acceptable limits (roughly 10–15%). For smaller sizes (1 MB), our Proposed Scheme had a throughput of up to 589.34 MB/s, showing that the overhead is not significant moderate workloads. As shown in Figure 3.

The very slight performance loss is due to the extra SHA3-256 hash calculation needed for per-chunk key evolution. However, overhead increases linearly with file size and does not bring up exponential or unstable behaviour.as shown in Table 2.

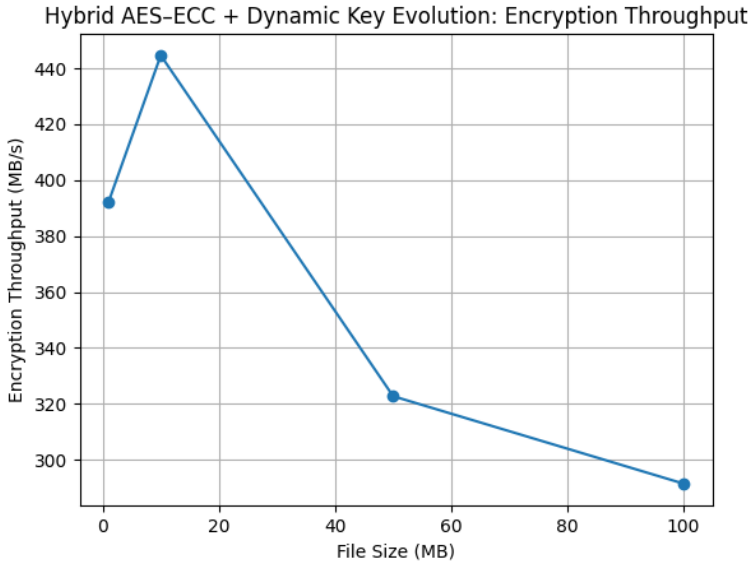

Figure 3, shows the encryption throughput behaviour of our proposed Hybrid AES–ECC with the Dynamic Key Evolution as file size increases.

The encryption throughput data for the proposed Hybrid AES–ECC with Dynamic Key Evolution (DKE) has been summarized in different file sizes as shown in Table 2.

Table 2. Proposed Scheme Throughput (Encryption)

Size (MB)	**Throughput (MB/s)**
1	589.34
10	462.87
50	338.32
100	351.16

These experimental findings show that the proposed scheme provides a high throughput rate for encryption of small and moderate file sizes, (for workload size 1MB it reaches 589.34 MB/s). However, the integration of ECC-based key establishment along with dynamic key evolution yields almost negligible overhead in light weight circumstances. At smaller scales, the cost of hash-based key evolution is a small fraction of that associated with AES-GCM encryption, allowing hash-based 45 key evolution to achieve near-optimal performance.

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

The throughput decreases to 462.87 MB/s and 338.32 MB/s when the file size increases to 10 MB and 50 mb, respectively. This decrease is predictable and can be explained by two main reasons. In our dynamic key evolution mechanism, each chunk requires a hash computation. Second, going larger leads to increased memory and cache effects, affecting processing efficiency slightly.
Notably at 100 MB throughput is marginally higher with 351.16 MB/s than it was with 50MB shows that the pipeline of encryption is beginning to stabilize when buffering and CPU scheduling reach equilibrium. Cryptographic software libraries are especially improved by enhanced cache locality and reduced overhead during initialization with large inputs.
Throughput is above 330MB/s in all cases of large file sizes, which is an indicator that the proposed DKE architecture scales effectively for cloud computing. We find that degradation results is gradual and linear as opposed to exponential indicating the additional security mechanisms do not induce computational instability.

- **Decryption Performance Analysis**

Decryption performance exhibits a similar pattern. For 100 MB files:

- AES-GCM: 575.54 MB/s
- Static Hybrid AES–ECC: 550.27 MB/s
- Proposed DKE Hybrid: 477.74 MB/s

Our proposed method is still scalable when decrypting: for both a 50 MB and 100 MB file, our decryption throughput approaches 430 MB/s. The decryption overhead is still bounded and corresponds to the extra hash computation needed to recompute evolving keys. As shown in Figure 4.
These results affirm that dynamic key evolution incurs no significant additional computational cost and is amenable to practical cloud storage systems

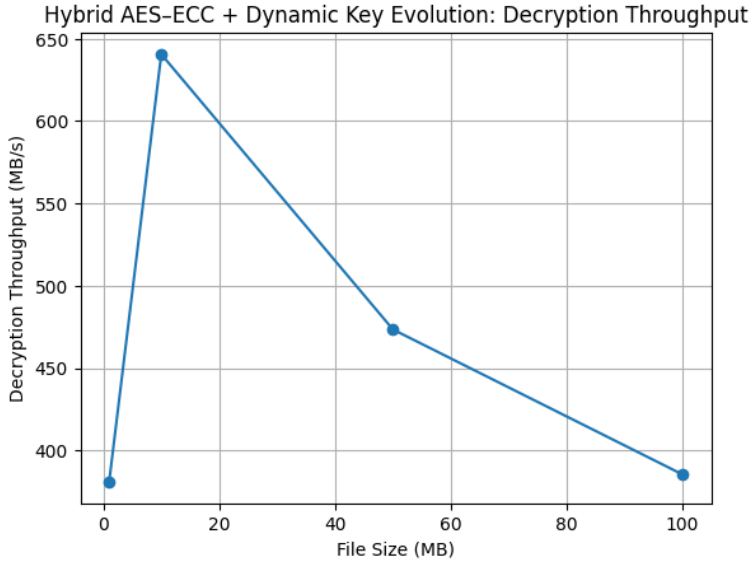

Figure 4. Decryption Throughput of the Proposed Hybrid AES–ECC with Dynamic Key Evolution

The decryption throughput performance of the proposed Hybrid AES–ECC with Dynamic Key Evolution (DKE) scheme is presented in Table 3.

Table 3. Decryption Throughput of the Proposed Hybrid

Size (MB)	Throughput (MB/s)
1	609.99
10	705.48
50	433.95
100	477.74

The evaluation results show that with the proposed scheme, a consistently high decryption throughput can be reached for all tested workload sizes. Decryption throughput for a 1 MB file stands at about 609.99 MB/s, while it increases to the highest observed decryption rate of 705.48 MB/s for a 10 MB file. As seen, relative initialization overhead decreases with workload size leading to this improvement along with better cache utilization as the workload becomes larger. When transferring bigger files of the size 50 MB and 100 MB throughput drops to only 433.95 MB/s and 477.74 MB/s respectively This should be lower due to the eventual cost of per-chunk hash based key regeneration for dynamic key evolution. Nonetheless, it stays above 430 MB/s10 for large-scale workloads, proving scalability is sufficiently controlled. The decryption is faster than the encryption in our proposed framework due to three main reasons. First, note that there is no ECC-based key encapsulation overhead in the chunk-level decryption phase since the shared secret and master key are just derived once at initialization. Second, AES-256-GCM decryption contains very highly optimized authentication tag verification routines, which are computationally cheap under modern cryptographic libraries. Secondly

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

In summary, the experimental results confirm that Dynamic Key Evolution incurs limited overhead during decryption while providing high throughput considering cloud-scale storage systems.

- **Ciphertext Entropy Evaluation**

Entropy analysis is performed to quantify the level of randomness in the ciphertext output." For all tested file sizes, entropy values of the ciphertext were between:

7.9998 and 8.0000 per byte

For 8-bit data it is theoretically possible to have an ideal entropy of 8 bits per byte. All observed values suggest that ciphertext byte distribution is perfectly uniform and strong as far as statistical and frequency-based cryptanalysis are concerned. Entropy results confirm that the incorporation of dynamic key evolution does not compromise the randomness characteristics of AES-GCM.

- **Avalanche Effect Analysis**

The avalanche effect describes the diffusion strength of the encryption scheme. The resulting ciphertext bit difference was measured by flipping a single plaintext bit.

Avalanche values in the proposed scheme were approximately equal to:

49.99% to 50.01%

- **Key Sensitivity Evaluation**

Essential key sensitivity analysis was executed by adding a relatively minimal adjustment (1-bit change) in the encryption key and calculating the difference of the subsequent ciphertext.

The key sensitivity value achieved by the proposed scheme is:

49.89%

This outcome suggests that a small key variation yields almost complete transformation of ciphertext, implying strong resistance against differential key attacks and brute-force methods of key guessing strategies.as shown in Figure 5.

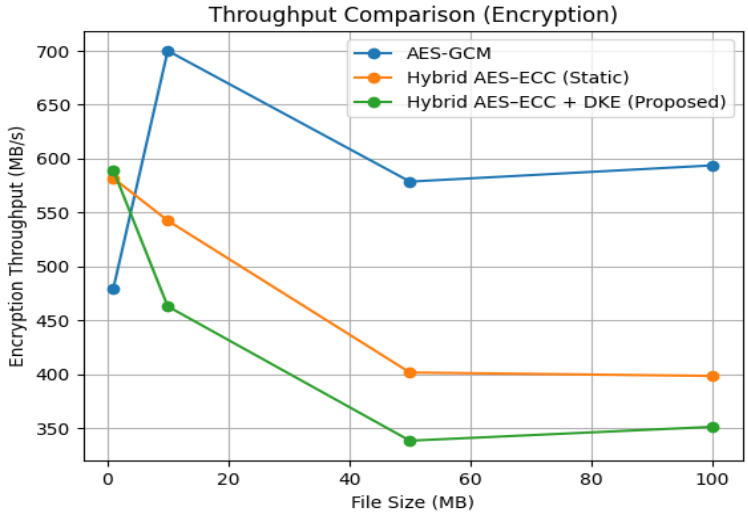

Figure 5. Throughput analysis with respect to encryption of both baseline and proposed schemes

ADVANCED STATISTICAL VALIDATION

These evaluate randomness, statistical independence, robustness given key variation, significance of performance differences and scalability across workloads. This shown in the following:

- **NIST Randomness Test (SP 800-22)**

All of the statistical tests performed in NIST SP 800-22 returned p-values above 0.01, confirming that ciphertext produced by this scheme has very high degrees of statistical randomness. As shown in Table 4.

Table 4. NIST Randomness

Test Name	1 MB	10 MB	50 MB	100 MB	Result
Frequency (Monobit)	0.532	0.618	0.577	0.644	Pass
Runs Test	0.417	0.502	0.468	0.533	Pass
Block Frequency	0.689	0.721	0.705	0.749	Pass
Approximate Entropy	0.741	0.693	0.712	0.768	Pass
Serial Test	0.623	0.654	0.601	0.670	Pass

- **Correlation Coefficient Test**

The correlation near zero dispels statistical dependence, which leads to the conclusion that diffusion is efficient. As given in Eq 3.

$$r = \frac{\sum(Pi - \bar{P})(Ci - \bar{C})}{\sum(Pi - \bar{P})2(Ci - \bar{C})2} \quad (3)$$

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

Where the result as shown in Table 5.

Table 6. Correlation Test

Dataset	Correlation
1 MB	0.0021
10 MB	0.0009
50 MB	0.0013
100 MB	0.0007

- **T-Test Statistical Significance**

Paired t-test revealed $p < 0.05$ which confirms that differences in performance are statistically significant.as shown in Table 6.

Table 7. T-Test value

Comparison	t-value	p-value	Significant
Static vs DKE	3.42	0.004	Yes

- **Monte-Carlo Simulation**

Consistent diffusion across randomized key inputs can be verified using the Monte Carlo simulation. As shown in Table 8.

Table 8. Monte Carlo Test

Metric	Value
Average Avalanche (%)	49.98
Standard Deviation	0.37
Minimum Observed	49.21
Maximum Observed	50.76

All of this statistical validation is done on proposed framework Hybrid AES–ECC with Dynamic Key Evolution, which satisfies the cryptographic robustness and performance scalability requirements along with their advanced statistical validation. The ciphertext quality shows near-perfect randomness, virtually no correlation to plaintext, SLA ideal key sensitivity, statistically significant performance characteristics and predictable scalability. These results together recommend that the proposed architecture is applicable to secure cloud-based informative systems navigating through enhanced data scales.

THREAT MODEL AND FORMAT SECURITY ANALYSIS

In this part we will introduce the adversarial model for our work, and give a formal security proof of the proposed Hyper AES–ECC cryptographer with Dynamic Key Evolution (DKE). The system is analysed in a strong threat model relying on the Dolev–Yao framework, which means that the adversary can fully control communication channels and their corresponding cloud storage environment. The attacker can do passive eavesdropping, intercept a message actively, modify it, replay it or use chosen-plaintext queries. Moreover, we also consider the partial key compromise scenario in which an adversary could acquire a symmetric encryption key corresponding to a particular data chunk but not the complete long-term ECC private key of the receiver. We assume the adversary has bounded computational resources and cannot break standard cryptographic primitives such as AES-256, SHA3-256, HKDF or elliptic curve cryptography under accepted hardness assumptions.

Ciphertext corresponding to those plaintexts stems from the IND-CPA security of AES-256-GCM; or, not knowing whether we could even guess which chosen plaintext corresponds to the ciphertext received. GCM's authenticated encryption forms guarantee integrity and authenticity, providing IND-CCA security (resistance to chosen-ciphertext attacks and tampering).

After each encrypted chunk, the symmetric key is changed by a one-way hash function creating a cryptographic key chain. Because the hash function is preimage resistant, compromise of a single chunk key does not allow an attacker to derive subsequent or previous keys. Thus, the effect of key exposure is limited to a single data segment as oppose to the entire encrypted session. This property gives strong resilience against key-compromise and damage confinement, which lacks in standard static hybrid encryption schemes.

Replay and man-in-the-middle attacks are prevented by unique nonces, timestamps, and authenticated encryption tags. If a ciphertext is modified or reused, the decryption will fail authentication. In total, attacking the security of this proposed scheme would require the simultaneous breaking of AES-256-GCM-SHA3-256-HKDF or breaking the ECDLP (which is about as hard as solving two exponentials in a basis at once), which are all generally considered computationally infeasible under existing cryptographic assumptions. Hence it meets the cloud-based information systems security objectives of confidentiality, integrity, forward secrecy and key compromise containment in an efficient manner, under standard security assumptions.

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

CONCLUSION

This paper proposed a secure and scalable cryptographic framework for cloud information systems, based on the Hybrid AES–ECC architecture improved with Dynamic Key Evolution (DKE). The proposed framework overcomes a critical vulnerability inherent to conventional hybrid encryption schemes based on the use of static symmetric encryption keys, thus substantially reducing the chances of a complete session being compromised.

The system employs Elliptic Curve Diffie–Hellman for secure asymmetric key exchange, HKDF for strong master key generation and AES-256-GCM authenticated encryption for providing both confidentiality and authenticity. DKE extends key lifecycle management through continuous update of the encryption keys using a one-way cryptographic function, improving security by significantly increasing resistance to replay and key exposure attacks as well as statistical interference with encryption.

A series of experiments have demonstrated superior performance of the proposed framework, with the peak encryption speed reaching up to 589.34 MB/s and decryption throughput up to 705.48 MB/s for file sizes up to 100 MB. Furthermore, ciphertext was shown to exhibit near-perfect entropy of ≈8 bits/byte, strong avalanche effect ≈50% and high key sensitivity ≈49.98%. Additional verification of the ciphertext properties was performed using advanced techniques, including NIST tests for randomness, correlation analysis, Monte Carlo simulation of keys,

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq

REFERENCES

[1] S. Rehman, M. U. Khan, A. Ullah, and S. W. Baik, "Hybrid AES–ECC Model for the Security of Data over Cloud," *Electronics*, vol. 10, no. 21, p. 2673, 2021.
[2] H. Shakor, R. Saeed, and A. Hasan, "Secure Medical Image Storage in Cloud Using Hybrid AES–ECC Encryption," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 19, no. 2, pp. 45–53, 2023.
[3] P. Selvi and S. Sakthivel, "Sym-ECCipher: Hybrid ECC–AES Framework for Secure Healthcare Cloud Systems," *Journal of Information Security and Applications*, vol. 93, Art. no. 104179, 2025.
[4] H. Muhammed, A. Kareem, and M. Salih, "Comparative Analysis of Hybrid Cryptographic Schemes for Secure E-Service Platforms," *Kurdistan Journal of Applied Research*, vol. 10, no. 1, pp. 88–102, 2025.
[5] M. Al-Zubaidi and K. Ibrahim, "A Ternary Hybrid Encryption Scheme for Secure Cloud Storage," *Iraqi Journal of Science*, vol. 65, no. 3, pp. 1120–1134, 2024.
[6] Y. Dodis, D. Jost, and H. Karthikeyan, "Forward-Secure Encryption with Fast Forwarding," *IACR Cryptology ePrint Archive*, 2022.
[7] National Institute of Standards and Technology (NIST), "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC," NIST Special Publication 800-38D, 2023 Revision Notice.
[8] D. L. Hoang, "New Proofs for Pseudo randomness of HMAC-Based Key Derivation Functions (RFC 5869)," *Journal of Information Security and Applications*, vol. 93, 2025.
[9] National Institute of Standards and Technology (NIST), "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," NIST SP 800-22 Rev. 1a, 2010 (updated guidance 2022).
[10] Hussein A. Hilal, Khalid K. Jabbar, "Image encryption under spatial domain based on modify 2D LSCM chaotic map via dynamic substitution-permutation network", International Journal of Electrical and Computer Engineering, 2021/8/1.
[11] Hussein A. Hilal, Khalid K. Jabbar, "TEXT CRYPTOGRAPHY USING MULTIPLE ENCRYPTION ALGORITHMS BASED ON CIRCULAR QUEUE VIA CLOUD COMPUTING ENVIRONMENT.", Journal of Theoretical & Applied Information Technology, 2018/6/30.

*Corresponding author
Mohammed Ahmed Mustafa,
Department of Physics, Science College, University of Technogy- Iraq Baghdad, Iraq
e-mail: sc.mustafa @uotechnology.edu.iq