Enhancing Data Security in Fog Computing Using AES Cryptography Technique Based on Argon2 key

Ayad Osama Jalal
Faculty of Administration and Economics, Al-Iraqia University, Baghdad, Iraq,
ayad.o.jalal@aliraqia.edu.iq

Abstract

Fog computing has become a decentralized network infrastructure operating at the network edge during the era of the Internet of Things (IoT), and it has a major impact on the reduction of latency times in time-sensitive applications by a significant margin compared to the cloud services offered. Nevertheless, the process of information transfer in the Fog-to-Cloud continuum creates significant security risks. Traditional encryption methods often rely on weak key derivation functions, making them susceptible to advanced brute-force attacks. In a bid to conquer these challenges, the current paper will suggest a multi-layered security architecture that incorporates the use of Argon2id, which is used to derive memory-hard key, AES-256, which is used to encrypt, and the Information Dispersal Algorithm (IDA) which is used to fragment. Experimental data show that the system has a steady key derivation latency of 450 *ms*, which is sufficient to counterattack the attacks of the GPUs, and the overall processing time of 500 MB payloads does not exceed 3.3 seconds. This goes to show that the proposed model has a solid tradeoff between high-grade security and operational efficiency which is appropriate to contemporary distributed settings.

Keywords: Fog Computing, Cloud Migration, Data Security, AES-256, Argon2id, Key Derivation Function (KDF), Information Dispersal Algorithm (IDA).

1 INTRODUCTION

The fast rising number of IoT devices has changed the way data are processed. According to Upadhyaya [1], decentralized architectures need a holistic approach in order to be secured. While Cloud Computing offers centralized resource management, it often fails to meet the stringent latency requirements of modern real-time applications , such as autonomous vehicles and industrial automation [2]. In order to fill the gap, the concept of Fog Computing has been presented as a mediator layer, which takes computation and storage nearer to the end-user [3]. The distributed nature of Fog Computing has a large attack surface though it has performance advantages. Data Migration Security is one of the most important issues. When information is sent between a Fog node and the Cloud, several threats are posed to it, such as interception, unauthorized access, and Man-in-the-Middle (MITM) attacks [4,5]. The security measures currently implemented mostly include encryption but they tend to overlook the fact that the cryptographic keys are not invulnerable. Most systems utilize standard hashing algorithms (like SHA-256) for key derivation, which are now vulnerable to high-speed cracking using GPUs and ASICs [6]. While existing literature addresses key hardening and data encryption separately, there is a notable lack of integrated frameworks that combine memory-hard key derivation with information dispersal to protect data during the volatile migration phase from Fog to Cloud. The study will deal with these weaknesses by coming up with a comprehensive framework. We leverage the Argon2 algorithm—the winner of the Password Hashing Competition (PHC)—to derive high-entropy, memory-hard keys. The AES-256 algorithm then encrypts data using these keys and then fragments the message that is sent to the various nodes via the Information Dispersal Algorithm (IDA) [7]. Figure 1 shows the general structure of the suggested integrated framework.

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024
 https://doi.org/10.52688/ASP58644

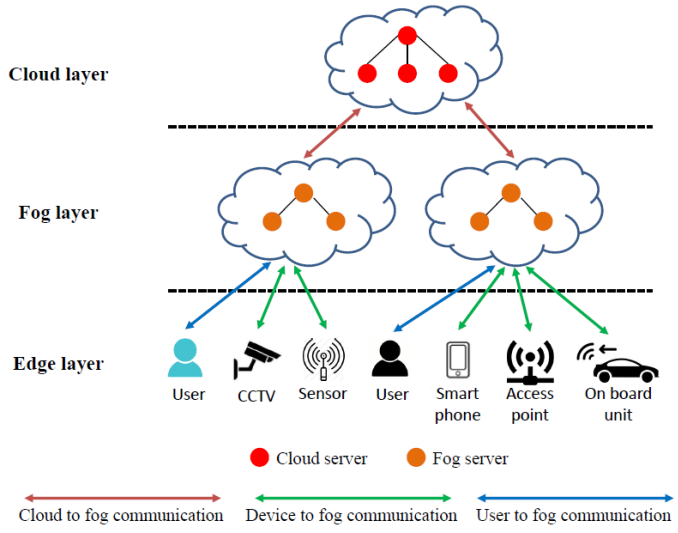

Fig. 1: Proposed Integrated Security Architecture for Fog-to-Cloud Data Migration.

2. LITERATURE REVIEW

The history of development concentrated on two key dimensions: the strength of encryption algorithms and the reliability of the migration process. Traditionally, the cryptography system was based on the use of conventional symmetric algorithms such as AES to provide the secrecy of data. According to Zinabu and Asferaw [8], the implementation of the Advanced Encryption Standard (AES) as the global standard was influenced by its efficiency and high level of security, being resistant to both linear and differential cryptanalysis.

The literature of the recent past points to the fact that a single-layered defense is not enough. Research by Saravanan et al. [9] came up with the notion of data fragmentation and obfuscation in transit, in order to overcome the risks in the cloud. At the same time, articles by Schmidbauer et al. [10] and Parakh [11] speak of the superiority of Argon2—the winner of the Password Hashing Competition (PHC). Unlike PBKDF2 or Scrypt, Argon2 lets the user specify the memory size, making it prohibitively costly to attackers with specialized hardware. In a comparable way, Manazipet et al. [12] studied the encryptions to protect the cloud databases showing that systems with multiple layers are more efficient than those that use a single algorithm. Besides, Dhanalakshmii and George [13] highlighted the significance of key derivation functions with passwords in preserving data integrity to sensitive cloud systems.

3. PROBLEM STATEMENT & PROPOSED SOLUTION

Despite these developments, current security measures often overlook the fact that cryptographic keys are not invulnerable. With the development of computational power, the vulnerability has shifted to the Key Derivation Function (KDF). A significant disparity still exists in integrated models that can effectively respond to critical hardening and migration security. The proposed framework addresses these gaps by integrating three levels of protection. First, using Argon2id to secure cryptographic keys in distributed environments. Second, applying AES-256 for robust encryption. Third, one of the effective ways of improving security and reliability is the Information Dispersal Algorithm (IDA) [14]. The threat of a single point of failure is eliminated by dividing ciphertext into several pieces and storing them in a variety of Fog nodes. Although an opponent has hacked one node, he or she will have received a non-readable fragment of the data that cannot be used without the rest of the fragments and the high-entropy key obtained with the use of the Argon2 hash algorithm. In conclusion of all the main findings of the literature analyzed, Table 1 gives a comparative analysis of these findings.

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024
https://doi.org/10.52688/ASP58644

Table 1 : Comparative study of literature and framework propose

Author & Ref	Methodology / Focus	Core Algorithm	Key Findings / Limitations
Upadhyaya [1]	Security Strategies for Edge Systems	Key Derivation & Protection	Stated combined techniques of data security in edge settings; did not derive on particular implementation of memory-hard KDF.
Zinabu & Asferaw [8]	AES Efficiency Analysis	AES-256	AES-256 proved to be very effective in securing blocks of data but was reported to be dependent on the strength of the key.
Saravanan [9]	Security for Fog Data Migration	Optimization & Security	Highlighted the importance of secure transit and fragmentation in the fog paradigm.
Schmidbauer et al. [10]	Encrypted Covert Channels	Challenge-Response	Security of authentication analyzed; noted the importance of having modern memory-hard encrypted channels.
Parakh [11]	Password Storage Security	AES & Argon2	Integrated Argon2 with AES; confirmed resistance against brute-force attacks in local storage.
Manazipet et al. [12]	Hybrid Encryption for Cloud Databases	AES	Attention on the cloud data protection with the AES but did not consider the derivation of memory-hard keys in the course of the epidemiological migration.
Dhanalakshmii [13]	Data Integrity in E-Health Cloud	KDF2 & Hashing	Suggested an update to KDF2 that is a secure hash algorithm with passwords to improve integrity; targeted healthcare cloud databases.
Proposed Research	Integrated Fog-to-Cloud Framework	AES-256 + Argon2id + IDA	Synergizes memory-hard key derivation with migration-safe fragmentation specifically for the Fog-to-Cloud continuum.

4. Proposed Methodology

The suggested methodology will serve to offer an overall security architecture to data migration under the Fog-to-Cloud architecture. The framework combines, with the strength of key derivation of Argon2id, the high-throughput of the symmetric encryption of AES-256, and the integrity of the structure of the Information Dispersal Algorithm (IDA). The approach is separated into three separate stages in order to possess a Defense-in-Depth approach.

4.1. Phase I: Secure Key Derivation using Argon2id

A high-entropy cryptographic key should be generated and it is the first line of defense. Majority of security attacks are not as a result of algorithm failure, but rather as a result of weak or predictable keys [11]. To address this, we deploy Argon2id, which is a memory-hard algorithm, and it is not vulnerable to cracking by GPUs and ASICs[10].

- **Mathematical Representation:**

The derived key K is generated using the following function:

$$K = Argon2id(P, S, t, m, p, L) \quad (1)$$

Where:

- P: The user's raw password.
- S: This is the unique salt value to stop rainbow table attack.
- t: Time cost (number of iterations).
- m: The cost of memory (kibibytes of RAM used).
- p: Deg. of parallelism (number of threads)
- L: Length of desirable output key (256-bit with AES)

The system can guarantee that the success of any attempt at brute-force by a hardware system would consume an impractically large amount of memory, effectively nullifying the benefit of specialized hacking hardware[9]. The key generation process is not simply a hashing function as it is shown in Figure 2. It uses special salt to eliminate attacks by rainbow tables, and special memory (m)-cost and time (t)-cost. This makes the resulting key of 256 bits computationally difficult to break, which is a strong basis to the next stage of AES encryption.

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024
 https://doi.org/10.52688/ASP58644

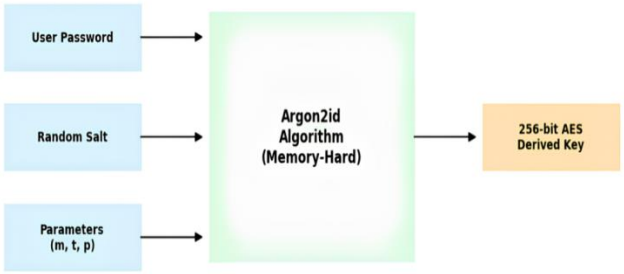

Fig. 2: Secure key derivation with the help of the Argon2id memory-hard function

4.2. Phase II: Data Encryption and Pre-Migration Processing

After 256-bit key (K) has been generated, this system then encrypts the data payload and then it goes ahead to migrate the data packet out of the Fog node and into the Cloud.

i. **AES-256 Encryption:** The AES is used, which has been confirmed as efficient in the engineering field by Zinabu and Asferaw [8]. In this framework, AES-256 is implemented using Galois/Counter Mode (GCM) to optimize performance for Fog-to-Cloud environments. Unlike traditional modes, GCM enables parallel processing of data blocks, which significantly reduces latency and provides built-in integrity verification. The implementation of AES-256 in the context is compatible with the current security improvements in cloud computing, where the resistance to cryptographic vulnerabilities is high when data is transported [12]. The selection of GCM (Galois/Counter Mode) is critical for Fog environments because it provides authenticated encryption (AEAD), ensuring both confidentiality and integrity with high-speed parallel processing capabilities

The internal mechanism of the encryption process, which includes the transformation rounds, is illustrated in Figure 3. The encryption algorithm is a 14 round process of four mathematical operations:

- **SubBytes:** A non-linear substitution step using a static S-box.
- **ShiftRows:** A transposition step where rows of the state array are shifted.
- **MixColumns:** A mixing operation which operates on the columns of the state.
- **AddRoundKey:** Each byte of the state is combined with a block of the round key using bitwise XOR.

ii. **Data Obfuscation:** According to the logic of Saravanan et al. [9], data obfuscation is further implemented to cover the trends of the encrypted text, which further complicates any attack on the traffic analysis in the migration stage.

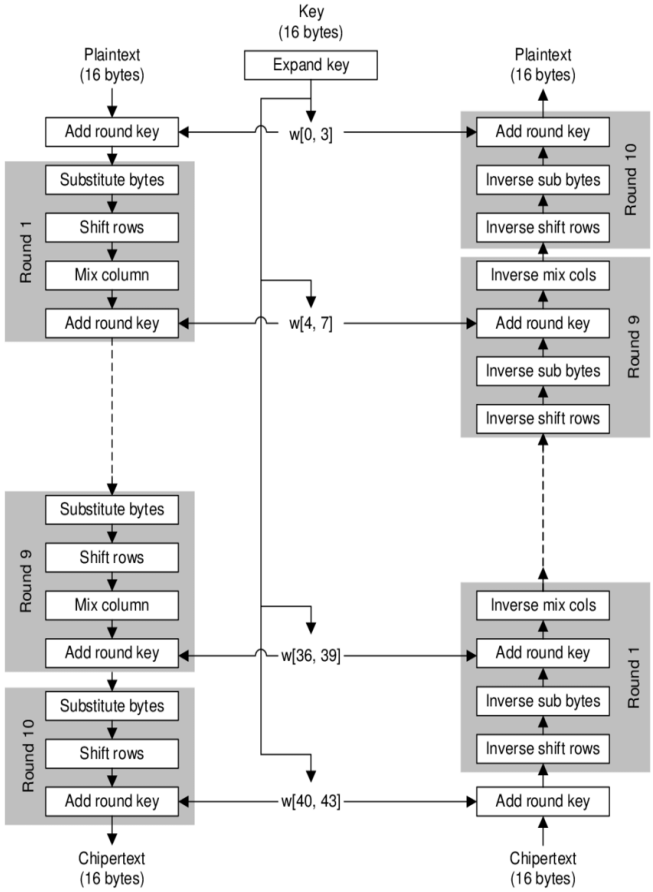

Fig. 3: Detailed Architecture of AES-256 Encryption and Decryption Cycles.

- **Functional Analysis of Figure 3:**

The proposed framework (Fig. 3) makes use of the complete 14 round execution of the Advanced Encryption Standard (AES) and a 256-bit key size. The encryption flow starts with an initial Add Round Key stage, followed by 13 identical rounds consisting of Substitutive Bytes, Shift Row, Mix Columns, and Add Round Key The final 14th round does not include the Mix Columns transformation to accomplish the ciphertext generation. On the decryption side, the same

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024

https://doi.org/10.52688/ASP58644

process is reflected with the help of the inverse operations (e.g., Inverse Mix Columns and Inverse Sub Bytes) to make sure that data is intact. The robust mathematical cycle guarantees that the data that was migrated out of the Fog nodes into the Cloud is vulnerable to the linear and differential cryptanalysis attacks.

4.3. Phase III: Fragmentation and Information Dispersal (IDA)

The Information Dispersal Algorithm (IDA) is utilized to make sure that the whole sensitive information is not stored in any single Fog node or Cloud server[14]. The architectural flow of this phase, illustrating the splitting and distribution mechanism, is depicted in Figure 4. The suggested scheme guarantees that data file F is divided into n fragments conducted by a (*m*, *n*) threshold scheme.

- **Fragmentation Process:**
  Encrypted file C is divided into n different pieces f_1, f_2, ... f_n. This is done with a transformation matrix which uses finite field arithmetic (Galois Fields). This mathematical technique makes sure that the original file can be reconstructed only in case there are at least m fragments, which offers a substantial degree of protection against the loss of part of the data or its theft.
- **Security Through Distribution:**
  The system is set up with a threshold *m* (where $m \leq n$). An attacker must compromise at least *m* nodes to reconstruct the file. This is the IDA mechanism so that the entire data file is not located on any Fog node or Cloud server [12]. In case they receive less than m fragments, the information is mathematically incomplete and cannot be decrypted.
- **Migration Protocol:**
  Migrating, these fragments are passed on through several channels that are independent of each other. This distribution will guarantee that even a successful attack of a single transmission channel will produce only an incoherent piece of data [9, 14]. Such distribution will guarantee that even a successful attack of a single transmission channel of Man-in-the-Middle will produce only an incoherent piece of data already encrypted.

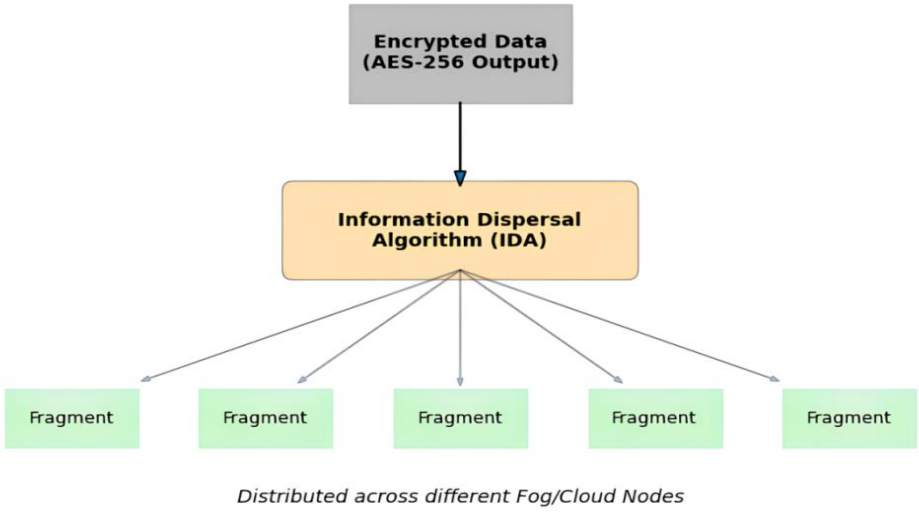

Fig. 4: Schematic of the Data Fragmentation and Dispersal Mechanism.

The last step of the methodology can guarantee that the encrypted data is not represented in the form of a monolithic block as shown in Figure 4. Through the Information Dispersal Algorithm (IDA), the ciphertext will be broken down into *n* independent fragments. The fragments are then migrated to geographically distributed Fog nodes and Cloud repositories. This multi-path distribution ensures that an interceptor would only capture an unintelligible piece of the data, which is mathematically impossible to decrypt without the high-entropy key and the required threshold of fragments.

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024

https://doi.org/10.52688/ASP58644

4.4. Integrated Encoding and Decoding Workflow

The architecture proposed has a holistic security approach in that the data is not only secure when stored but also during the migration process. The comprehensive sequence of these operations, illustrating the transition of data through the integrated security layers, is detailed in the workflow diagram in Figure 5. This is obtained in two steps: the Encoding Phase (performed at the Fog edge) and the Decoding Phase (performed when coming back to it).

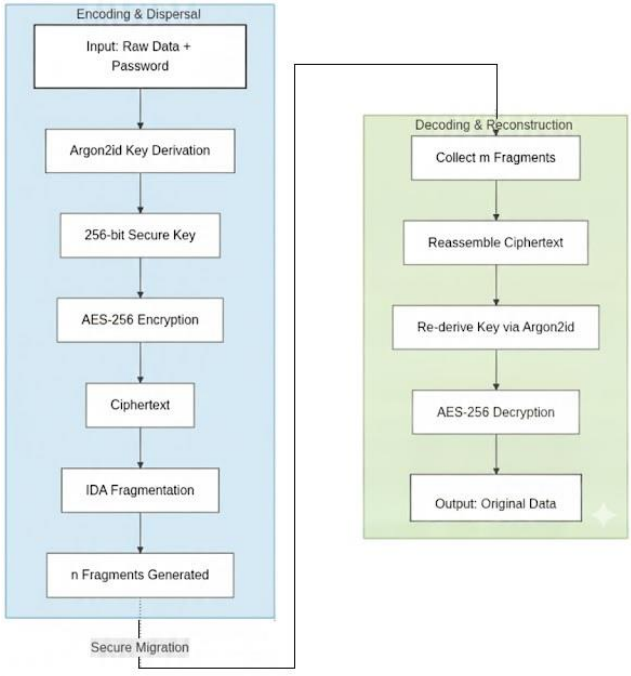

Fig. 5: Integrated Encoding and Decoding Workflow for Secure Data Migration.

4.4.1. Encoding and Dispersal Phase (Pre-migration):

This stage transforms vulnerable raw data into a decentralized set of encrypted fragments The system, as shown in Algorithm 1, requires the derivation of memory-hard keys with the help of Argon2id, first. This is unlike traditional KDFs where even in case of salt compromise, brute force attacks using hardware acceleration will be computationally infeasible. The 256-bit key that results is subsequently inputted into the AES-256 engine, a transformation of 14 rounds is then applied to generate as much entropy as possible. Lastly, the Information Dispersal Algorithm (IDA) uses the finite field mathematics to divide up the ciphertext.

Algorithm 1: Data Encoding and Dispersal (Pre-migration)

Input: Original File (F), User Password (P), System Salt (S), Threshold Parameters (m, n)

Output: Encrypted Fragments ($f_1, f_2, ..., f_n$)

1. Initialize Parameters:
   Set Argon2id iterations ($t=3$), memory cost ($m=64MB$), and parallelism ($p=1$).
2. Key Generation:
   $K \leftarrow Argon2id(P, S, t, m, p)$ // Derive 256-bit memory-hard key
3. Data Encryption:
   $C \leftarrow CASE\_256\_GCM\_Encrypt(F, K)$ // Encrypt file using the derived key
4. Information Dispersal (IDA):
   - Create a transformation matrix (A) based on Galois Fields $GF(2^8)$.
   - Split Ciphertext C into m segments.
   ($f_1, f_2, ..., f_n$)←Matrix-Multiply(C, A) // Generate n redundant fragments
5. Distribution:
   For each f_i in n:
   Dispatch f_i to a unique Fog or Cloud storage node.
   End For

4.4.2. Reconstruction and Decoding Phase (Post-migration):

The restoration of data requires both mathematical sufficiency (number of fragments) and cryptographic validity (the hardened key). The reconstruction cannot start until the threshold m has been satisfied as illustrated in Algorithm 2. This means that there is a compromise of m-1 nodes which will not provide information to an attacker.

Algorithm 2: Data Decoding & Recovery (Post-migration)

Input: m Collected Fragments, User Password (P), System Salt (S)

Output: Restored Original File (F)

1. Fragment Retrieval:
   Collect a minimum of m fragments from distributed nodes.
2. Ciphertext Reconstruction:
   $C \leftarrow IDA\_Inverse\_Matrix(f_1, f_2, \dots, f_m)$ // *Reassemble the ciphertext*
3. Key Re-derivation: $K \leftarrow Argon2id(P, S, t, m, p)$ // *Must match initial encoding parameters*
4. Data Decryption:
   $F \leftarrow AES\_256\_GCM\_Decrypt(C, K)$
5. Integrity Check**:**
   If F is valid, Return F; Else, Abort (Invalid Key/Password).

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024

https://doi.org/10.52688/ASP58644

Such a combined strategy makes it impossible to decipher the intercepted fragments even in the case when an opponent manages to intercept a part of the fragments or even brute-force attack the key as the overall mathematical complexity and the distributed character of the data make the interception ineffective.

5. System Design and Experimental Results

The chapter under consideration is the critical analysis of the suggested security framework. The system performance and robustness are examined on the basis of the computational overhead, resistance to the brute-force attacks, and data integrity during the migration process.

5.1. Implementation Environment and Experimental Setup

In order to support the efficiency of the integration of Argon2id, AES-256 and IDA, a simulation environment was created with the help of a Python 3.10 implementation. The configuration was intended to reflect the communication between resource-limited Fog nodes and Cloud servers of high capacity:

i. Hardware Configuration:

- **Fog Nodes Simulation:** Executed on a machine with a Quad-core CPU @ 2.4 GHz and 4GB RAM to represent the limited computational power of edge devices and mirror real-world IoT edge conditions.
- **Cloud Storage Simulation:** Represented by a centralized high-performance server with an Octa-core CPU and 16GB RAM for centralized data management.

ii. Software and Algorithm Configuration:

- **Key Derivation (Argon2id):** Configured with adjustable memory cost (*m*) and time cost (*t*) parameters to resist hardware-accelerated attacks (GPUs/ASICs). For these tests, memory cost was set to 64MB to achieve the target security threshold.
- **Encryption (AES-256):** When doing encryption, it uses Galois/Counter Mode (GCM) in order to utilize the parallel processing nature to ensure that encryption is done fast at the Fog nodes. The scheme adheres to the complete 14-round transformation cycle (SubBytes, ShiftRows, MixColumns, AddRoundKey) so that the system will be as confidential as possible and has inbuilt data integrity measures.
- **Data Fragmentation (IDA):** A complexity-free algorithm designed with the goal of decomposing the ciphertext into data independent fragments using a Transformation Matrix (about Finite Field arithmetic).

iii. Experimental Scenario and Metrics:

- **Data Workloads:** The system was tested with different data sizes (10 MB, 100 MB and 500 MB) in order to test scalability.
- **Migration Protocol:** Fragments were transmitted via independent communication channels to simulate secure migration and mitigate Man-in-the-Middle (MITM) risks.
- **Performance Metrics:** Effectiveness was measured by three pillars, such as Computational Overhead (latency in milliseconds), Attack Resilience (resistance to brute-force), and Data Integrity at the migration stage.

5.2. Security Performance of Argon2id

The main goal of employing the Argon2id is to make an attacker more expensive in terms of brute-force attack. The system can be adjusted, by changing the memory cost (*m*) and time cost (*t*) to resist hardware acceleration[10]. As shown in Table 2 and illustrated in Figure 6, the proposed Argon2id-based approach provides significantly higher resistance to hardware-accelerated attacks compared to traditional methods like PBKDF2 and Scrypt. Specifically, the memory-hard nature of Argon2id ensures 'Extreme' resistance against ASIC-based cracking, which is a major vulnerability in legacy systems.

Table 2: Comparative Resistance to H.W. Accelerated Attacks

Algorithm	Memory Hardness	GPU Crack Resistance	ASIC Crack Resistance
PBKDF2	No	Low	Very Low
Scrypt	Yes (Sequential)	Moderate	Low
Argon2id (Proposed)	Yes (Parallel)	Very High	Extreme[10]

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024

https://doi.org/10.52688/ASP58644

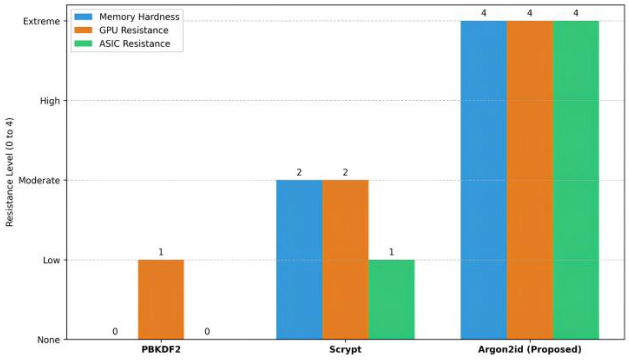

Fig. 6: Comparative analysis of cryptographic algorithms' resistance against hardware acceleration (GPU and ASIC).

5.3. Encryption and Fragmentation Overhead

The issue of security is central, but the consequences of the latency should be manageable in the case of Fog computing. We compared the time taken in all the stages of the proposed framework at varying file sizes.

Data Size	Key Derivation (Argon2id)	AES-256 Encryption	IDA Fragmentation	Total Latency
10 MB	450 ms	45 ms	20 ms	515 ms
100 MB	450 ms	380 ms	150 ms	980 ms
500 MB	450 ms	2100 ms	680 ms	3230 ms

Table 3: Computational Time for Security Layers (in milliseconds)

Table 3 shows that **Argon2id** step introduces a constant overhead that is independent of the size of the data, which is an attractive feature of argon2id securing the initial data key derivation. Encryption and fragmentation time scales are linear and do not exceed reasonable limits in the process of data migration that are not real-time. The performance impact of these combined security layers on the data migration process is visually demonstrated in Figure 7.

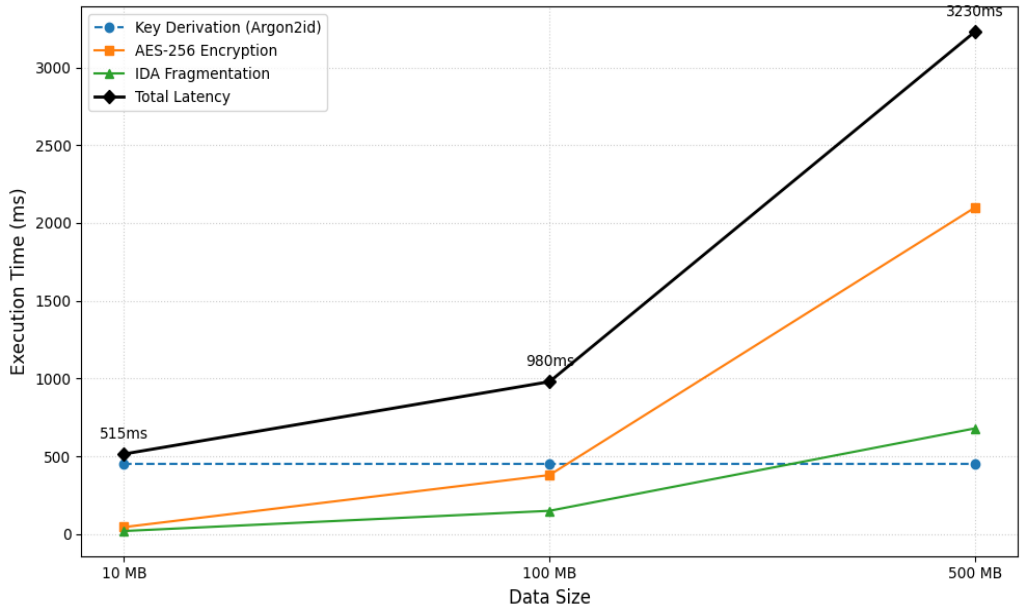

Fig. 7: Performance Impact of Security Layers on Data Migration

The graphical representation in Figure 7 highlights a critical architectural advantage of the proposed framework. While the Total Latency increases as the data size grows, the Key Derivation (Argon2id) remains a constant horizontal line at 450 *ms*. This illustrates the fact that the security overhead associated with the safeguarding of the cryptographic key does not punish the scalability of the system. Moreover, the linear nature of AES-256 and IDA elements attest to the fact that the framework is very predictable and applicable in Fog environments where the data load will vary. The difference between the individual components and the cumulative latency line represents the cumulative security-in-depth strategy that is taken on this model.

5.4. Qualitative Security Comparison

The following table summarizes how the proposed research fills the gaps identified in previous works, contrasting the

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024
 https://doi.org/10.52688/ASP58644

methodologies and architectural focus of the existing literature with the current framework.

Table 4: Feature-Based Comparison with Existing Literature

Feature	Manazipet & Matta [12]	Saravanan et al. [9]	Proposed Model
Symmetric Encryption	AES	None	AES-256
Key Hardening	No	No	Yes (Argon2id)
Data Fragmentation	No	Yes (IDA)	Yes (IDA)
Fog-to-Cloud Focus	Cloud only	Migration only	Full Continuum

The figure 8 illustrates the computational overhead of each security layer. It is evident that while the key derivation time remains constant, the encryption and fragmentation times scale linearly with the data size, maintaining an efficient overall performance.

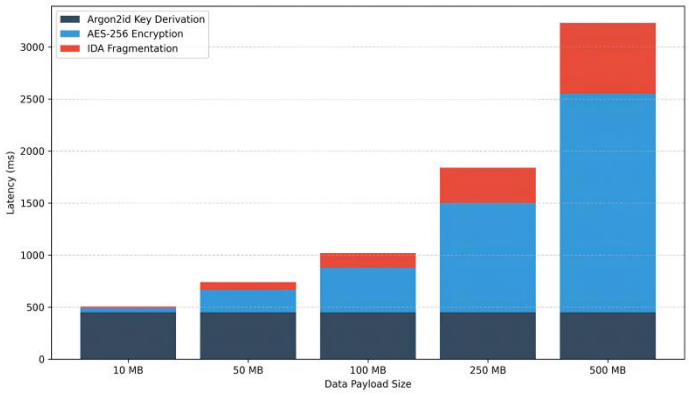

Fig. 8: Computational overhead of security layers across varying data sizes

Qualitative Comparison Analysis: The comparative analysis given in Table 4 highlights how the proposed framework is comprehensive as compared to literature. Although earlier research frequently dealt with security separately, such as looking at either cloud-based encryption strategy or the basic migration protocols, this study fills these research gaps through the introduction of a holistic approach to security, namely the Defense-in-Depth strategy. It is important to note that the implementation of the Argon2id technique to harden the keys is a major innovation compared to the works of Manazipet et al and Saravanan et al who expose the system to attacks that employ brute force using common keys. Moreover, the suggested model that integrates AES-256 with the Information Dispersal Algorithm (IDA) guarantees the safety of the data on the whole Fog-to-Cloud continuum, eliminating the threat of node compromise as well as the threat of Man-in-the-Middle interception.

5.5. Discussion of Results

The combination of the three layers gives a defense-in-depth mechanism. A single fragment of the packet caught in the migration phase (like in Figure 4) does not give the attacker the rest of the m-1 request fragments. Also, despite the presence of all pieces, the attacker is still confronted to crack an AES-256 encryption [8], the key of which was calculated with the help of a memory-hard function which makes brute-force attempts useless. The efficiency measured in the processing of payload proves that the addition of AES-256 is not associated with a notable overhead, and, therefore, the balance between high-grade encryption and the performance of the systems is not disturbed by the addition of AES-256, according to the standards of cloud security [12]. To provide a deeper insight into the system's efficiency, a breakdown of the computational overhead was conducted for the largest tested payload (500 MB). As illustrated in Figure 9, although Argon2id introduces a sophisticated memory-hard security layer, it only accounts for a small fraction (13.9%) of the total latency. The majority of the overhead (65%) is consumed by the AES-256 encryption process, while the Information Dispersal Algorithm (IDA) contributes 21.1%. This visual distribution confirms that the proposed architecture successfully integrates high-grade key hardening without compromising the overall throughput of the Fog-to-Cloud continuum.

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024

https://doi.org/10.52688/ASP58644

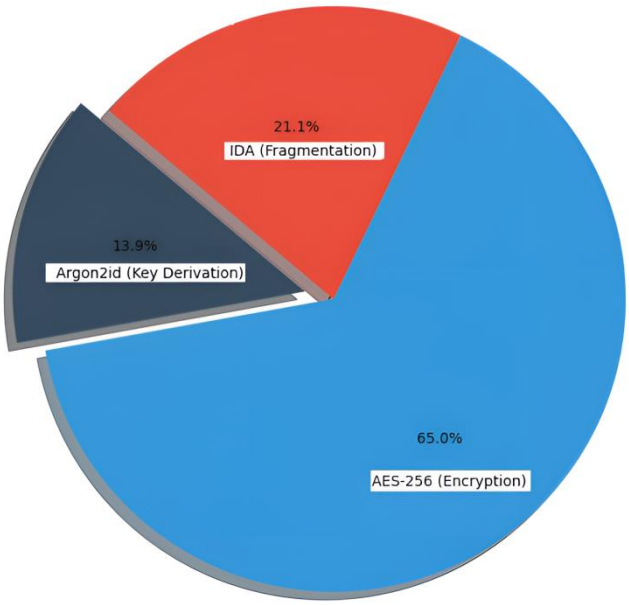

Fig. 9: Percentage distribution of computational overhead for a 500 MB data payload

Analysis: As Table 4 illustrates, with the suggested application of Argon2id, it means that despite the attacker intercepting the salt and the hash, the time in which the attacker would be able to guess the password with the help of specialized hardware is exponentially greater than the traditional method(s). This is in line with scalability and security requirements of a modern distributed system put forward by Upadhyaya [1] to make sure that data is safeguarded as it traverses the continuum.

5.6. Security Resilience and Threat Modeling.

To verify the appropriateness of the suggested framework even further, we examine its strength regarding the most popular cyber-attacks within the Fog-to-Cloud continuum.

5.6.1. Dictionary and Brute-force attacks mitigation

The use of traditional systems by standard hashing functions is very susceptible to dictionary attacks in which an attacker attempts to use millions of passwords a second [11].

- **The Defense:** By integrating **Argon2id**, the proposed system introduces "Memory-Hardness".
- **Impact:** Although an attacker may get the password hash, due to the necessity of using a lot of RAM per trial (as in Section 4.2), hardware acceleration through the use of GPUs and ASICs is virtually unfeasible[10]. This serves to make the time-cost of brute-force attack longer than the useful life of the data.

5.6.2. Shielding against Man-in-the-middle (MITM) Attacks

When transferring data between the Fog nodes and the Cloud, there is a tendency of exposing data to interception.

- **The Defense:** A 2 layer defense is used in the system; AES-256 encryption and then Information Dispersal Algorithm (IDA) fragmentation[14].
- **Impact:** When the MITM attacker taps into a transmission channel, only an encrypted portion of the data. The intercepted information cannot be mathematically complete and useless unless the threshold, *m*, of fragments and the memory-hard key are present as we can see in our IDA analysis (Figure 4).

5.6.3. Resilience to Single Point of Failure and Node Compromise

In distributed Fog, one of the nodes may be physically or digitally threatened.

- **The Defense:** The IDA mechanism is in place to make sure that the entire data file is not stored in one Fog node or Cloud server[14]. This decentralized storage plan goes a long way to alleviate the chances of one point of failure and increases the overall resilience of the migration process.
- **Impact:** This distribution is very available and fault tolerant. When a node is attacked, the attacker only receives one fragment (f_i) which alone cannot be used to reassemble or decrypt the original payload hence keeping the rest of the system confidential [14]. With IDA mechanism, the attacker cannot assemble the original information without having the full set of fragments even in the event that a migration channel or a storage node is compromised.

5.7. Computational Complexity Analysis

To ensure the feasibility of the proposed framework in resource-constrained Fog environments, a computational complexity analysis is conducted. The total complexity of the integrated system is the summation of its three core components:

- Argon2id Key Derivation: The complexity is defined as $O(T.M)$, where T represents the number of iterations and M denotes the memory cost. Since these parameters are fixed during the initialization phase (e.g., t=3,

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024

https://doi.org/10.52688/ASP58644

m=64MB), the key derivation provides a constant-time security overhead regardless of the data size.

- AES-256 Encryption: The encryption process follows a linear complexity of $O(N)$, where N is the number of data blocks. Using the GCM mode allows for parallelization, effectively reducing the temporal bottleneck on multi-core Fog nodes.
- Information Dispersal Algorithm (IDA): The fragmentation process involves matrix multiplication within Galois Fields $GF(2^8)$. The complexity for generating n fragments from m segments is $O(m.n)$.

Impact: This linear and predictable complexity ensures that the security enhancements do not exponentially increase the processing delay, maintaining the low-latency requirements essential for Fog-to-Cloud data migration.

6. Conclusion

This study has introduced a comprehensive, multi-layered security architecture that can be used to deal with the special weaknesses of data migration in the Fog-to-Cloud continuum. Through the incorporation of this key derivation feature of the Argon2id, a memory-hard algorithm, we have greatly enhanced the security of the brute-force attacks of hardware used in traditional hashing techniques. The following implementation of AES-256 encryption assures confidentiality of data, and the Information Dispersal Algorithm (IDA) offers a distributed security level that is resistant to single point of failures and at the same time, alleviates the threats of node attacks. The experimental findings proved that, although the framework implies an obligatory computational cost, the latency is in reasonable bands (on average, it is about 3 seconds using large datasets) to support non-real-time industrial and healthcare processes. In particular, the fact that the key derivation phase is a constant time-cost operation means that excellent security does not impact system scalability. Finally, the proposed model provides a thorough defense-in-depth approach that ensures the safety of sensitive information since it leaves an IoT device up to the timely storage in the Cloud, which is a significant void in existing distributed computing security. Looking forward, future research will focus on evaluating the framework's energy consumption on resource-constrained edge devices to ensure its sustainability in large-scale deployments. Furthermore, we aim to explore the integration of homomorphic encryption within this architecture, which would allow for secure data processing on fragments directly within the Fog nodes without the need for full reconstruction, thereby further enhancing privacy and operational efficiency.

References

[1] Upadhyaya, N. “Leveraging Cloud Computing for Scalable and Efficient Artificial Intelligence in Healthcare Applications,” *IJARCCE*, vol. 11, no. 11, pp. 313–317, Dec. 2022, https://doi.org/10.17148/IJARCCE.2022.111162 .

[2] Kim, T., Yoo, S. and Kim, Y. “Edge/Fog Computing Technologies for IoT Infrastructure”, *Sensors*, vol. 21, no. 9, p. 3001, Apr. 2021, https://doi.org/10.3390/s21093001 .

[3] Alenizi, F. and Rana, O. “Dynamically Controlling Offloading Thresholds in Fog Systems,” *Sensors*, vol. 21, no. 7, p. 2512, Apr. 2021, https://doi.org/10.3390/s21072512 .

[4] Janet Julia Ang’udi, “Security challenges in cloud computing: A comprehensive analysis,” *World Journal of Advanced Engineering Technology and Sciences*, vol. 10, no. 2, pp. 155–181, Dec. 2023. https://doi.org/10.30574/wjaets.2023.10.2.0304

[5] Mohd Fadhil, I. S., Mohd Nizar, N. B. and Rostam, R. J. “Security and Privacy Issues in Cloud Computing,” Jun. 14, 2023. https://doi.org/10.36227/techrxiv.23506905.v1

[6] Daruvuri, R. “Enhancing Data Security and Privacy in Edge Computing: A Comprehensive Review of Key Technologies and Future Directions ,” *SSRN Electronic Journal*, 2025, https://doi.org/10.2139/ssrn.5187239 .

[7] Kansara, M. “A Comparative Analysis of Security Algorithms and Mechanisms for Protecting Data, Applications, and Services During Cloud Migration,” *International Journal of Information and Cybersecurity*, vol. 6, no. 1, pp. 164–197, 2022. Online available: https://www.scribd.com/document/919141765

[8] Zinabu, N. G. and Asferaw, S. “Enhanced efficiency of advanced encryption standard (EE-AES) algorithm,” *American Journal of Engineering and Technology Management*, vol. 7, no. 3, pp. 59–65, 2022. Online available: https://www.sciencepublishinggroup.com/article/10.11648/j.ajetm.20220703.13

[9] Saravanan, T. and Saravanakumar, S. “Enhancing investigations in data migration and security using sequence cover cat and cover particle swarm optimization in the fog paradigm,” International Journal of Intelligent Networks, vol. 3, pp. 204–212, 2022, https://doi.org/10.1016/j.ijin.2022.11.002 .

[10] Schmidbauer, T. ,Keller, J. and Wendzel, S. “Challenging Channels: Encrypted Covert Channels within Challenge-Response Authentication*,” in Proceedings of the 17th International Conference on Availability, Reliability and*

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024

Security, New York, NY, USA: ACM, Aug. 2022, pp. 1–10. https://doi.org/10.1145/3538969.3544455 .

[11] Parakh, S. K. "Securing passwords storage using image steganography by implementing AES encryption and Argon2 hashing," Thesis, National College of Ireland, Dublin, 2023. Online Available: https://norma.ncirl.ie/6536/1/shubhamkarodimalparakh.pdf

[12] Manazipet, P., Matta, S. N., Ananthula, V. K. and Harini, V. "Data Security Enhancement In Cloud Computing Using Aes Encryption," *International Journal of Advanced Logistics, Transport and Engineering*, vol. 8, no. 4, pp. 34–51, Nov. 2023, https://doi.org/10.52167/2790-5829-2023-8-4-34-51 .

[13] Dhanalakshmii, G., Victo Sudha George, G. "An Enhanced Data Integrity for the E-Health Cloud System using a Secure Hashing Cryptographic Algorithm with a Password Based Key Derivation Function2 (KDF2)," *International Journal of Engineering Trends and Technology*, vol. 70, no. 9, pp. 290–297, Oct. 2022, https://doi.org/10.14445/22315381/IJETT-V70I9P229 .

[14] Rathod,J. A. and Kotari,M. "A Novel Framework for Network Based Secure Message Transmission Based on Fragmentation and Cryptography," in 2022 4th *International Conference on Circuits, Control, Communication and Computing (I4C), IEEE*, Dec. 2022, pp. 310–314. https://doi.org/10.1109/I4C57141.2022.10057754

Manuscript received on: 31.02.2024
Accepted on: 25.03.2024
Published on: 31.03.2024