Comprehensive Journal of Science
Volume (10), Issue (38), (Dec, 2025)
ISSN: 3014-6266

مجلة العلوم الشاملة
المجلد (10) ملحق العدد 38 ديسمبر 2025
ردمد: 3014-6266

Comparative Performance Study on Symmetric and Asymmetric Encryption Algorithm: Comprehensive Report of a Comparative Study

Amel Abdyssalam A Alhaag

Faculty of Information Technology, University of Az-Zawiya

Email: am.alhaag@zu.edu.ly

تاريخ الاستلام:2025/12/8 –تاريخ المراجعة: 2025/12/12– تاريخ القبول: 2025/12/19– تاريخ للنشر : 17/ 1/ 2026

الملخص

إن الطفرة في الاتصالات الرقمية و التجارة الإلكترونية و أنظمة الحوسبة السحابية سلطت الضوء على أهمية أنظمة التشفير لضمان أمن تبادل المعلومات. تُعد خوارزميات التشفير المتماثلة و غير المتماثلة من أكثر نماذج التشفير دراسةً، و تُشكل اللبنات الأساسية لأنظمة الأمان الحالية. و على الرغم من أن كلا النوعين يوفران نفس الدرجة من السرية، إلا أن خصائص أدائهما و أدائهما الفعلي يختلفان اختلافًا كبيرًا. ستُقارن هذه الورقة البحثية بين نوعي الخوارزميات مع التركيز على سرعة الحوسبة، متطلبات الذاكرة، قابلية التوسع و مقاومة الهجمات. أظهرت النتائج التجريبية و الدراسات السابقة أن الخوارزميات المتماثلة مثل معيار التشفير المتقدم أكثر كفاءة و سرعة بينما من ناحية أخرى تُعدّ الخوارزميات غير المتماثلة بما في ذلك خوارزمية ريفست شامير أدلمان و خوارزميات تشفير المنحنى الإهليلجي غير فعّالة حسابيًا إلا أنها تُقدّم مزايا في توزيع المفاتيح و عمليات المصادقة, تُعزى هذه النتائج إلى الإتجاه المتزايد نحو النظر في أنظمة التشفير الهجينة التي تجمع بين نقاط قوة كلا النهجين لتحقيق التوازن بين الأداء والأمان (ستارلينغ, 2017; سينغ و كومر 2020).

Abstract

The boom in the digital communication, e-commerce and cloud-based systems has highlighted the nature of cryptographic systems in the ensuring of security in information exchange. Symmetric and asymmetric encryption algorithms belong to the most studied cryptographic paradigms and the building blocks of the existing security systems. Although both types provide the same degree of confidentiality the performance features of the two types and their actual performance is very different. The present paper will provide a comparison of the two kinds of algorithms with a focus on the computational speed, memory requirement, scalability, and attack resistance. The experimental findings and literature have demonstrated that the symmetry algorithms such as Advanced Encryption Standard (AES) is more efficient and quick in its simultaneous performance and must be applied in the case of encrypting huge amounts of information. On the other hand, asymmetric algorithms, including the Rivest Shamir Adleman (RSA) and Elliptic Curve Cryptography (ECC) are computationally inefficient, yet offer advantages in distributing keys, as well as authentication. These findings are the reason why there is an ever-growing tendency to consider hybrid cryptosystems incorporating the strengths of both approaches to reach an equilibrium of performance and security (Stallings, 2017; Singh and Kumar, 2020).

Keywords: Symmetric encryption, Asymmetric encryption, Performance evaluation, AES, RSA, ECC, Cryptography

1. Introduction

The increased application of computer technologies in communication, financial systems and cloud computing has led to the increased need of the effective encryption mechanism. Encryption implies that data is confidential, intact and authentic, which are essential elements of a cybersecurity model (Katz and Lindell, 2020). Symmetric and asymmetric encryption algorithms are the two paradigms of cryptography that have their advantages and disadvantages. Symmetric algorithms such as AES and DES only need a single common key, are computationally simple and extremely fast. In contrast, symmetric algorithms operate on a pair of key (public and private), which solves the long-standing problem of key distribution and makes it possible to issue secure digital signatures (Menezes, Van Oorschot, and Vanstone, 2019). The study and the industry require the comparative study of symmetric and asymmetric encryption algorithms. Organizations are faced with trade-offs between efficiency, scalability and security. To use one as an example, symmetric encryption can encrypt large amounts of data very effectively, but does not offer suitable key management, whereas asymmetric encryption offers suitable key management, and can be slower at bulk encryption (Stinson and Paterson, 2019). The performance of such protocols as Transport Layer Security (TLS), where an asymmetric encryption is used on the exchange of session keys, and a symmetric encryption is used on the exchange of data, is affected by such trade-offs (Rescorla, 2018). The research question that will be adhered to in the current research is as follows: What is the comparison of symmetric and asymmetric encryption algorithms in terms of performance, scalability, and practical usage in the frames of the contemporary settings imposed on the modern computational environment? The aims of the research are:

1. To study theoretical foundations of symmetric and asymmetric encryption.

2. To compare the selected algorithms (AES, DES, RSA, ECC) concerning their execution time, the amount of memory, and scalability.

3. In order to evaluate the practical implications of each algorithm to the real world service e.g. cloud services, mobile computing and IoT setups.

4. To explore the hybrid systems of encryption that will combine the two paradigms. Towards these ends, the paper will be targeted at providing a comprehensive comparative report, which will help practitioners, researchers and engineers to make decisive cryptographic choices. The role that this paper plays in the general body of knowledge of performance trade-offs. Cryptographic design also comes in handy because it sheds light on the problem that are nevertheless very relevant in the presence of growing fears regarding cyberattacks and the imminent risk of quantum computing (Chen et al., 2016).

2. Literature Review

Symmetric and asymmetric encryption algorithms have been an important subject of the cryptographic research since decades. This part examines the theory, real-world application, and comparison works of the literature available. The review is divided into three sections, which are symmetric encryption, asymmetric encryption, and comparative research trends.

2.1 Symmetric Encryption Algorithms

Some of the oldest cryptography algorithms are the symmetric encryption algorithms, also referred to as the secret-key algorithms. They are safe due to the confidentiality of a common key to both obtain encryption and decryption. The primary advantage of symmetric encryption is that the identical key is used, and the computational expenses are low, and encryption and decryption can be done significantly quickly (Stallings, 2017). The Data Encryption Standard (DES) that the U.S. National Institute of Standards and Technology (NIST) standardized in 1977 is one of the earliest popular symmetric algorithms. DES took a key of 56 bits, which in turn proved insecure due to advances in the brute-force attack. Triple DES (3DES) and the improvement application was achieved through triple usage of DES with various keys. This increased security, but significantly decreased efficiency compared to more modern algorithms (Katz and Lindell, 2020). The new standard, AES that was implemented in 2001 came into place in the place of DES. AES is deployed on Rijndael cipher and it has the key sizes of 128, 192, and 256 bits. It has already been demonstrated that AES can withstand brute-force attacks pretty well and delivers high throughput rates on different platforms, including hardware-accelerated environments (Daemen and Rijmen, 2013). Moreover, the hardware manufacturers like Intel have made the AES-NI instructions available in their processors that provide colossal performance improvements (Gueron, 2012). Symmetric encryption is preferred by the real-time applications such as video streaming, database encryption, and wireless communication since it is fast and predictable in terms of memory usage (Singh and Kumar, 2020). The greatest weakness however is on key distribution: sharing a secret key safely with parties, especially in large scale systems is hard.

2.2 Asymmetric Encryption Algorithms

The cryptography or asymmetric encryption developed by mid-1970s was the creation to eliminate the weakness of symmetric encryption particularly the issue with distribution of keys. In an asymmetric algorithm, a key and another key are employed that are mathematically equivalent: one key is public key that is applied in encryption of the message and the second key is used in decryption of the message (Diffie and Hellman, 1976). This technology facilitated the transmission of safe messages and opened the door to the secure Internet communications without sharing secret key beforehand. The most conspicuous among asymmetric algorithms was Rivest -Shamir-Adleman (RSA) introduced in 1977. RSA relies on mathematical complexity of the factoring problem of large prime numbers, which at classical computers could be said to be computationally infeasible (Rivest, Shamir, and Adleman, 1978). RSA has been used in both protocols of secure socket layer (SSL) and in the use of transport layer security (TLS) to establish secure session. However, RSA has good security resilience at the cost of performance: key generation, encryption and decryption is computationally infeasible, especially with large key sizes (Stinson and Paterson, 2019). Another more efficient replacement to RSA has been ELLiptic Curve Cryptography (ECC). The basis of ECC is the algebraic properties of the elliptic curves over finite fields and provides a comparable or better level of security with much smaller key sizes. Using the example of a 256-bit ECC key, it is claimed to be as secure as 3072-bit RSA key (Menezes et al., 2019). The effectiveness of ECC predisposes it especially to low-resource devices, e.g., smartphones and Internet of Things sensors. There is clear trade-off between RSA and ECC as observed in numerous research works. Despite the relative simplicity that has made RSA still popular, ECC is gaining popularity in new applications due to the relatively small number of computation requirements (Koblitz and Menezes, 2015). Asymmetric algorithms also form the basis of digital signatures, certificate authorities, and blockchain technologies, and this is why they are required in authentication and the verification of integrity.

2.3 Comparative Research Trends

Symmetric and asymmetric algorithm comparative analysis reveals that there is an apparent trade off between speed and key management. Raw throughput and efficiency are always poorer, particularly of AES, in asymmetric algorithms compared to symmetric algorithms. However, the asymmetric algorithms are especially powerful when it comes to providing safe communication channels and organization of digital identities (Rescorla, 2018). A comparison of the AES, DES, RSA and ECC performance in different file sizes by Singh and Kumar (2020) found out that encrypting and decrypting data was much faster with symmetric algorithms. Comparatively, RSA and ECC were slower in computation but they had higher assurances of key security in exchange. Similarly, Kaur and Gupta (2019) also emphasized that hybrid solutions are indispensable since the use of asymmetric algorithms is impossible when a certain amount of data is to be encrypted, whereas it is invaluable in the context of setting the secure sessions. The advent of hybrid cryptosystems in which both symmetric and asymmetric encryption are combined has been a pointer that there is an ever growing consensus that each of these two families alone cannot support the full spectrum of security needs. During the TLS, e.g. asymmetric encryption defines the session key, and in the process of data transfer, the transfer is secured by symmetric encryption. Cloud computing, mobile application, and blockchain are developing new systems that continue to be built on this hybrid approach to deliver efficiency over robust security (Al-Bassam, 2018). Comprehensively, the literature proves the fact that symmetric encryption is faster, more significant in the speed of scale as compared to asymmetric encryption and in security, the latter provides the necessary qualities of exchanging keys and validation. These two paradigms continue to take part in the design of new cryptographic systems.

3. Methodology

The study uses a mixed-method design of theoretical review and experimental benchmarking. It is aimed at creating a multifaceted comparative analysis of symmetric and asymmetric encryption algorithms, their indicators of performance, scalability and applicability.

3.1 Research Design

The methodology includes two main stages:

1. **Literature-based synthesis:** A systematic review of prior studies was performed to identify the theoretical advantages, limitations, and common applications of symmetric and asymmetric algorithms. Peer-reviewed articles, cryptographic standards, and authoritative textbooks were consulted (Stallings, 2017; Katz & Lindell, 2020).
2. **Experimental benchmarking:** Algorithms were tested in a controlled computing environment to assess their real-world performance. Performance metrics included execution time, memory consumption, key generation overhead, and throughput across varying input sizes.

3.2 Algorithms Selected

Four widely used algorithms were selected to represent both categories:

- **Symmetric algorithms:** Advanced Encryption Standard (AES), Data Encryption Standard (DES).
- **Asymmetric algorithms:** Rivest–Shamir–Adleman (RSA), Elliptic Curve Cryptography (ECC).

These were chosen because AES is the most common symmetric standard, DES (though outdated) provides historical context, RSA is the most widely used asymmetric algorithm, and ECC represents modern efficiency improvements (Menezes et al., 2019).

3.3 Experimental Setup

The experimental tests were conducted on a system with the following configuration:

- **Processor:** Intel Core i7-11800H, 2.3 GHz
- **Memory:** 16 GB RAM
- **Operating System:** Windows 11 (64-bit)
- **Programming Environment:** Python 3.11 using the PyCryptodome and cryptography libraries

3.4 Performance Metrics

The following metrics were used to evaluate performance:

- **Encryption time (ms):** Time taken to encrypt data.
- **Decryption time (ms):** Time taken to decrypt data.
- **Throughput (MB/s):** Rate of encryption per second.
- **Key generation time (ms):** Time taken to create cryptographic keys.
- **Memory consumption (MB):** Peak memory usage during operations.

Data files ranging from 1 MB to 100 MB were used to simulate real-world scenarios such as database encryption, file storage, and secure communication.

3.5 Data Analysis

Experimental outcomes were averaged among ten runs of algorithm and file size to obtain statistical stability. All of the runs were performed at the same system conditions with background processes reduced to the minimum to minimize noise in measurements. The repetition depth of ten was important since cryptographic functions commonly interact with subsystem operating system processes, memory allocation policies and CPU scheduling. These temporal fluctuations were averaged by running the same experiment several times, which resulted in more valid averages (Stallings, 2017).

3.5.1 File Types and Testing Conditions

To simulate real world use cases, there were not only plain text documents in the test files but also images in the JPEG format and video samples in the MP4 format. The motivation behind the application of multimedia was that the encryption algorithms process the input data variable with varying degrees of entropy. Data of high-entropy, including compressed video, is usually less vulnerable to trends that can be used in cryptanalysis, yet can still indicate variations in throughput among algorithms (Katz and Lindell, 2020). AES was continuously recorded to be stable irrespective of the type of file but RSA was recorded to be variable especially with large video files.

3.5.2 Performance Metrics Beyond Speed

Although core performance indicators are the execution time and memory consumption, energy consumption was also taken into account in this study. Average encrypted and decrypted power draw of the CPU was measured by using the Intel Power Gadget tool. The symmetric algorithm like AES used much less energy per MB of data than RSA and this is expected of the higher level of computation of the RSA (Singh and Kumar, 2020). ECC, being more efficient than RSA, nevertheless used almost twice the energy per MB as AES. Such results are especially applicable to battery-operated devices such as smart phones and internet of things sensors.

3.5.3 Statistical Validity and Robustness

Each experiment had the standard deviation and variance calculated in order to determine stability. Both AES and DES exhibited low variance, which is generally less than 2-3 percentage points of the mean and this reassures uniformity. Variability in RSA results was more variable and standard deviations of some runs were greater than 15%. As a measure to validate differences, statistical testing was used:

* The tests of normalcy (Shapiro-Wilk) were used to confirm that the encryption times were distributed approximately normally.

* Significant differences between symmetric and asymmetric groups were confirmed with paired t-tests ($p < 0.01$). * Multivariate analysis of variance (MANOVA) was used in cases where time, memory and power consumption were compared at the same time. It was found that there was a statistically significant difference in the type of algorithms in terms of performance indicators ($d = 0.001$), which enhances the validity of conclusions (Stinson and Paterson, 2019).

3.5.4 Outlier Handling

Some anomalies were identified, especially when using RSA keys, certain runs were spiked to above 400 ms, as compared to their average of 250 ms. Outliers were checked and differently assigned to cache conflicts and background OS scheduling. Instead of reporting the mean value and the median value in isolation, both were provided to represent a balanced picture. These anomalies also showed a tendency to decrease median values, which proves the relative consistency of ECC over RSA (Kaur and Gupta, 2019).

3.5.5 Visualization of Results

The interpretation of the results was based on data visualization. Scalability was demonstrated by line charts where AESs demonstrated linear increases, DES slower but predictable increases, and RSA demonstrated exponential increases in the encryption times with increasing data sizes. • Evidence of Bar graphs was used to compare average power consumption among algorithms and it is clear that AES was the least power-intensive. Heat maps Cross-validation algorithms were cross-compared on three axes (time, memory, energy), which give a holistic view of efficiency. Scatter plots were prepared to indicate how key size correlates with the execution time. Indicatively, RSA-4096 was almost 7 times slower than RSA-2048 and ECC-521 only increased moderately relative to ECC-256.

3.5.6 Sample Results for Larger Files

Table 2 provides sample results for 10 MB and 50 MB files, which better reflect realistic workloads.

Table 2. Average Performance of Algorithms on Larger Files

Algorithm	File Size	Encryption Time	Decryption Time	Power Consumption (W)	Memory (MB)
AES-128	10 MB	45 ms	44 ms	8.2	14
AES-128	50 MB	226 ms	220 ms	8.7	14
RSA-2048	10 MB	14.2 s	13.8 s	24.5	65
RSA-2048	50 MB	71.3 s	70.9 s	27.1	68
ECC-256	10 MB	3.6 s	3.4 s	15.8	34

Algorithm	File Size	Encryption Time	Decryption Time	Power Consumption (W)	Memory (MB)
ECC-256	50 MB	17.9 s	17.2 s	16.3	35

As shown, AES maintained sub-second encryption times even for 50 MB files, while RSA became practically unusable for files larger than 10 MB. ECC performed better than RSA but still lagged far behind AES.

3.5.7 Integrated Interpretation

Such results point not only to the raw performance of algorithms, but to their applicability to real-world application. AES can provide close and constant efficiency regardless of the type and size of the data, whereas RSA and ECC are poorly scaled, and should be used to exchange keys, as opposed to bulk encryption. The combination of quantitative analysis, checking of variances, energy profiling, and visualization in the present study makes it a sound and multi-dimensional appreciation of algorithmic performance (Menezes et al., 2019).

3.5.8 Case Study: TLS Handshake and Session Encryption

Transport Layer Security (TLS) protocol is one of the most typical practical uses of encryption and it is used to encrypt Internet traffic, including web browsing and Internet banking. TLS is built on asymmetric and symmetric encryption in a hybrid design. In the process of the TLS hand shake, asymmetric key algorithms are utilized to exchange a common session key between the client and the server, including RSA or ECC (Rescorla, 2018). After this key is swapped, bulk data encryption is replaced by symmetric algorithm, which is often AES. In order to recreate this scenario, OpenSSL libraries were used to configure a test environment in order to measure handshake latency and session throughput: • RSA-2048 handshake: mean latency = 120 ms. • ECC-256 handshake: latency mean of 40 ms. • AES-128 session encryption throughput: >200 MB/s when the key has been set. These findings support the need to have hybrid systems. In the absence of asymmetric encryption, key exchange would not be safe across an untrusted network. However, with the absence of symmetric encryption, data communication would be computationally infeasible. The connection between the two paradigms justifies their co-existence in practically all secure communication paradigms (Singh and Kumar, 2020). This is due to the fact that most of the articles analyzed are published online on various platforms and thus can be accessed by the researcher online.<|human|>Multi-Platform Performance Evaluation

3.5.9 The majority of the articles reviewed are found online and, therefore, available to the researcher online. To investigate practical deployment, more tests were made in three environments:

1. Desktop- Performance: Intel i7 processor, 16 GB RAM. 2. Midrange smartphone: ARM cortex-A76 processor, 6 GB RAM. 3. IoT microcontroller ARM Cortex-M4, 256 KB RAM. Findings have shown the following trends: On the desktop platform, AES took less than 250 ms to encrypt a 50 MB file compared to RSA which took more than 70 seconds to do so. ECC was also quicker than RSA, but still was an order of magnitude slower than AES. On the smartphone, AES performance was also good, with the 50 MB encryption time being approximately 800 ms, whilst RSA was practically useless, taking several minutes. ECC provided acceptable performance (approximately 25 seconds), thus it was selected when secure communications are needed in a mobile setting (Kaur and Gupta, 2019). On the IoT device, AES was suitable in small data-sizes (less than 1 MB) and small delay, whereas RSA was impractical because of limited memory. However, ECC managed to accomplish key exchange in approximately 500 ms, which proves the appropriateness of the method in the limited environment (Menezes et al., 2019). These platform-perceptive insights point out the flexibility of symmetric algorithms to all environments and the special benefit of ECC to lightweight, embedded domains.

3.5.11 attack-resistance and security metrics. Performance measures prevail over experiment analysis but attack resilience is also to be considered. Symmetric algorithms like AES can normally withstand brute-force attack in case of key lengths long enough (128 bits or more). Nevertheless, they can be compromised when keys are not handled properly or even used again (Katz and Lindell, 2020). Asymmetric algorithms have the merit of authentication and key management with other threats. The use of integer factorization by RSA is endangered by developments in quantum computing, and the use of elliptic curve discrete logarithm problems by ECC is endangered in the same way (Chen et al., 2016). In order to capture these issues, the strength of security was quantified using bits of security whereby AES-128 corresponds to 128 bits of security, RSA-2048 corresponds to approximately 112 bits, and ECC-256 corresponds to approximately 128 bits. This adds to the strength of ECC in having a good security with less key size and high-speed compared to RSA (Bernstein et al., 2009). 3.5.11 Future-proofing Analysis. The conglomerate analysis indicates that performance and security can be considered in terms of technological development. Existing symmetric algorithms, especially the AES, are still resistant to predictable attacks, whereas the asymmetric algorithms are threatened by quantum computing. In this way, the current research in post-quantum cryptography is centered on lattice-based and code-based systems, which are supposed to offer quantum-resistant analogs but at sufficiently viable performance tiers (Bernstein et al., 2009). 3.5.12 Comparatively Understanding Enlarged Visualization. To enhance interpretability layered visualizations have been created: • Multi-dimensional comparisons of AES, DES, RSA and ECC were made using 3D plots of throughput, memory and energy consumption. Box plots were used to show the variance, and as it was revealed, RSA has an unstable pattern relative to the tight distribution of AES. Cumulative distribution graphs were used to permit probabilistic evaluation of encryption time, indicating that 95% AES executions took less than 5 ms with 1 MB files, and RSA took more than 1 second almost uniformly. These new visualization methods provided micro-level detail (algorithms stability) and macro-level detail (applicability in the real world).

4. Results

4.1 Symmetric Algorithm Performance

AES consistently outperformed DES in all experiments. On a 1 MB file, AES-128 achieved an encryption time of 4.6 ms, while DES required 9.2 ms. When file sizes increased to 100 MB, AES maintained linear scalability, completing encryption in approximately 450 ms, while DES took 870 ms. These results align with findings by Singh and Kumar (2020), who observed AES's superior throughput compared to legacy symmetric algorithms.

In terms of memory usage, AES averaged 12 MB across operations, while DES required 15 MB due to multiple rounds of processing. AES also demonstrated more predictable performance across increasing file sizes.

4.2 Asymmetric Algorithm Performance

RSA was significantly slower than ECC. On average, encrypting a 1 MB file with RSA-2048 took 1.5 seconds, compared to 0.38 seconds with ECC-256. For a 10 MB file, RSA encryption required over 13 seconds, making it impractical for large-scale data. By contrast, ECC completed the same task in less than 4 seconds.

Key generation also highlighted performance differences: RSA-2048 required an average of 250 ms for key generation, while ECC-256 completed in 65 ms. These findings align with Koblitz and Menezes (2015), who emphasized ECC's efficiency in modern environments.

Memory consumption further reinforced ECC's advantage. RSA used an average of 60 MB of memory, while ECC required only 30 MB.

4.3 Comparative Summary

A comparative overview of symmetric and asymmetric algorithms is provided in Table 1.

Table 1. Performance Comparison of Selected Algorithms

Algorithm	Type	Avg. Encryption Time (1 MB)	Key Generation Time	Throughput (MB/s)	Memory Usage (MB)
AES-128	Symmetric	4.6 ms	N/A	215	12
DES	Symmetric	9.2 ms	N/A	108	15
RSA-2048	Asymmetric	1.5 s	250 ms	0.65	60
ECC-256	Asymmetric	0.38 s	65 ms	2.63	30

These results confirm that symmetric algorithms dominate in terms of speed and efficiency, while asymmetric algorithms are indispensable for secure key management and digital identity (Rescorla, 2018).

4.4 Key Observations

1. **Speed:** AES outperformed all algorithms, while RSA was the slowest.
2. **Scalability:** Symmetric algorithms scaled linearly with data size, while asymmetric algorithms scaled poorly.
3. **Key Management:** Only asymmetric algorithms offered secure key distribution, with ECC being more efficient than RSA.
4. **Practical Suitability:** AES is ideal for bulk data encryption; ECC is preferable for secure communications in resource-constrained environments.

---

5. Discussion

The experimental findings, coupled with insights from existing literature, shed light on the performance trade-offs and application contexts of symmetric and asymmetric encryption algorithms. This section interprets the results, connects them to real-world applications, and explores the implications for hybrid cryptosystems.

5.1 Speed and Efficiency

The results clearly show that symmetric algorithms, especially AES, outperform asymmetric algorithms in terms of raw speed and computational efficiency. AES completed encryption tasks several orders of magnitude faster than RSA, making it highly suitable for encrypting bulk data such as multimedia files, cloud storage, and real-time communication (Daemen & Rijmen, 2013). These findings are consistent with earlier studies that highlighted AES as the de facto standard for efficient encryption (Stallings, 2017).

Asymmetric algorithms, by contrast, displayed significant computational overhead. RSA's performance was particularly limited, with encryption times extending into seconds for relatively small datasets. This confirms the widely held view that RSA is impractical for bulk encryption (Stinson & Paterson, 2019). ECC, however, demonstrated improved performance while maintaining high levels of security, reinforcing its growing adoption in resource-constrained environments such as IoT and mobile devices (Koblitz & Menezes, 2015).

5.2. Scalability and Resource Usage. Symmetric algorithms were also linear in their behavior with respect to growth in data size and their encryption and decryption times were predictable. The fact that they do not consume much memory is also an added advantage in making them more applicable in large-scale data systems, including database encryption and distributed file systems (Singh and Kumar, 2020). By contrast, asymmetric algorithms did not scale very well and performance reduced significantly when files became larger. The memory usage and turnaround times of RSA increased proportionately and therefore it cannot be used in applications where the encryption is necessary to maintain high speed. ECC offered a more balanced alternative that has reduced memory consumption and quicker operations when compared to RSA, which validated its ability as the asymmetric technique of choice in future applications (Menezes et al., 2019). 5.3 Security Considerations Cryptographic evaluation does not stop at performance, but in terms of security strength, it is most relevant. Symmetric algorithms are based on secret keys making them vulnerable to key distribution. In case of secret key breach, the whole communication is jeopardized (Katz and Lindell, 2020). Asymmetric algorithms overcome this shortcoming through the use of the public-private key pairs whereby the keys can be distributed over insecure channels (Diffie & Hellman, 1976). Moreover, digital signatures are facilitated by asymmetric cryptography which offer authentication and non-repudiation. This feature is essential in e-commerce, online banking, and blockchain (Al-Bassam, 2018). Therefore, although the asymmetric algorithms are less fast, they are still a necessity. 5.4 Practical Applications The results show symmetric algorithms are best suited to: • Encryption of bulk data (databases, files, media). • Video streaming (VoIP, communication in real time). Embedded systems with predictable performance needed. Asymmetric algorithms, and the ECC in particular, are more appropriate to: • Encrypt the keys in TLS and VPN. Authentication and digital signatures. • Mobile and IoT applications in which key management is required. It is the combination of these strengths that has become the defining feature of the current cryptographic architectures; hybrid encryption systems. To give an example, asymmetric encryption defines a secure session key in TLS, and then clear bulk data encrypting occurs through a symmetric encryption (Rescorla, 2018).5.5 Emerging Challenges: Post-Quantum Cryptography

A major factor deserving future research is the emergence of quantum computing. By efficiently computing discrete logarithm and factoring large primes, Shorer asserts that the algorithm will threaten to break RSA and ECC (Chen et al., 2016). Symmetric algorithm like the AES is not as susceptible to quantum attacks but it could still take less time to ensure that its key length is not compromised. This has prompted the development of studies on post-quantum cryptographic algorithms, which aim to deliver quantum-resistant security but at performance-reasonable levels (Bernstein et al., 2009).

6. Conclusion

This paper was a detailed comparative experiment on the symmetric and asymmetric encryption algorithms that integrated both literature review and experimental benchmarking. Key conclusions include: 1. Performance: Symmetric algorithms, especially AES are much faster, scalable, and less memory consuming than asymmetric algorithms and are well suited to bulk encryption. 2. Key Management: Asymmetric algorithms are slower, but they can be used to offer secure key distribution and authentication mechanisms that are the cornerstones of secure communications. 3. Practical Applications: AES is still the most popular option with data intensive applications and ECC is becoming popular with authentication and secure communication under limited environments. 4. Hybrid Techniques: The best strategy is a combination of symmetric and asymmetric encryption, as the most popular protocols such as TLS can prove. 5. Perspectives, The emergence of quantum computing brings new challenges, such that quantum-resistant algorithms should be studied to provide long-term cryptographic security. To sum up, it is not that symmetric and asymmetric encryption are competitors but rather that they are complementary paradigms. Their integration is the basis of the modern secure systems with performance and solid security. The future of digital ecosystems will be, and always will be, essential in ongoing research of hybrid models and post-quantum cryptography.

References

Al-Bassam, M. (2018). Blockchain-based decentralized cloud computing. *IEEE Cloud Computing, 5*(4), 36–42. https://doi.org/10.1109/MCC.2018.043221659

Bernstein, D. J., Buchmann, J., & Dahmen, E. (2009). *Post-Quantum Cryptography*. Springer.

Chen, L. K., Jordan, S., Liu, Y. K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). *Report on post-quantum cryptography*. National Institute of Standards and Technology.

Daemen, J., & Rijmen, V. (2013). *The design of Rijndael: AES — The Advanced Encryption Standard*. Springer Science & Business Media.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory, 22*(6), 644–654. https://doi.org/10.1109/TIT.1976.1055638

Gueron, S. (2012). Intel Advanced Encryption Standard (AES) New Instructions Set. *IEEE Security & Privacy, 7*(1), 64–69.

Katz, J., & Lindell, Y. (2020). *Introduction to Modern Cryptography* (3rd ed.). CRC Press.

Kaur, A., & Gupta, R. (2019). Comparative analysis of RSA and ECC for resource-constrained devices. *International Journal of Computer Applications, 178*(39), 1–7.

Koblitz, N., & Menezes, A. (2015). The random oracle model: A twenty-year retrospective. *Designs, Codes and Cryptography, 77*(2-3), 587–610.

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (2019). *Handbook of Applied Cryptography*. CRC Press.

Rescorla, E. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3*. Internet Engineering Task Force.

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM, 21*(2), 120–126. https://doi.org/10.1145/359340.359342

Singh, S., & Kumar, R. (2020). Performance analysis of symmetric and asymmetric cryptography algorithms for security. *Journal of Computer Science, 16*(5), 689–699.

Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson.

Stinson, D. R., & Paterson, M. B. (2019). *Cryptography: Theory and Practice* (4th ed.). CRC Press.