NTU Journal of Engineering and Technology (ICSDT2025) special issue
DOI: https://doi.org/10.56286/ntujet.icsdt2025

IRAQI Academic Scientific Journals

P-ISSN: 2788-9971 E-ISSN: 2788-998X
NTU Journal of Engineering and Technology
Available online at: https://journals.ntu.edu.iq/index.php/NTU-JET/index

Performance Analysis of Network Efficiency Based on Multi-Level Encryption Algorithms

Heba Aldabagh[1], Omar Ibrahim Alsaif[2], Younis Anas Younis[2]
[1] Engineering Technical College/Mosul, Northern Technical University/Mosul/Iraq, Northern
[2]Technical University technical Engineering College for Computer and AI, Northern Technical University/Mosul/Iraq,
heba_aldabagh93@ntu.edu.iq, omar.alsaif@ntu.edu.iq, younis.alrozz@ntu.edu.iq.

Article Informations

Received: 15-03- 2025,
Accepted: 28-04-2025,
Published online: 01-03-2026

Corresponding author:
Name: Omar Ibrahim Alsaif
Affiliation: Northern Technical University
Email: omar.alsaif@ntu.edu.iq

Key Words:
Multi-Level encryption, cryptographic algorithms, performance analysis, encryption speed, decryption speed.

A B S T R A C T

With an increasing rate of data exchange via networks rapidly, information security is a critical indicator of network effectiveness. Data transmission has to be secured, and that is crucially done through cryptography, where AES, DES, and multi-level encryption (AES+DES) are the significant methods. This research provides a comparative performance analysis of AES and DES. Execution times for encryption and decryption, efficiency ratios, and memory usage were recorded using various dataset sizes (1, 10, 100, 1,000, 10,000, and 50,000 images). Results show that AES is faster than DES in all cases with less memory. Experiments were conducted on both CPU and GPU: Results indicate that GPU acceleration does make a difference in accelerating encryption with up to a (6.68×) speedup for multi-level encryption and (8.13×) for AES with 50,000 images. GPU-memory usage was (35%) less than the respective CPU-based memory, thus being more efficient. Multi-level encryption presents a potential trade-off for more robust security, while DES is insufficient for bulk encryption because its performance is considerably lowered.

1. Introduction

Because of the many communications today, where large amounts of information are exchanged over networks and between different countries, one of the criteria for measuring the efficiency of networks is to ensure the security of information transmitted over the network through what is known as network security [1]. Network security provides four security services: confidentiality, integrity, authentication, and nonrepudiation at the message level and another authentication service at the entity level [2].

One of the methods and techniques to ensure network security is cryptography, which is the science of hiding information and making it unreadable to anyone except the person concerned with that information, who is the only one able to return the unreadable text to the original text to obtain information from it. To prevent intruders and untrustworthy people from obtaining information, the original text was encrypted with one of the encryption algorithms and with a key. The receiving party decrypts the cipher text using an encryption algorithm and a key. The type of encryption algorithms depends on the type of key used. If the same key is used in the encryption and decryption process, that is called symmetric encryption, where the key used is the secret key. However, if a public key is used on the sender's side to encrypt the original text and a private key is used on the receiver's side to decrypt the cipher text and return it to the original text, the encryption algorithms used are of the asymmetric encryption type [3]. Encryption plays a crucial role in securing digital communications. AES and DES are both symmetric key encryption algorithms, but they differ significantly in terms of security, performance, and computational complexity, especially for large-scale data encryption [4]. This study explores the efficiency of AES, DES, and AES + DES encryption across CPU and GPU architectures, focusing on execution time and memory usage. The goal is to determine the optimal approach for balancing security and computational efficiency. Among the most widely used encryption algorithms are AES (Advanced Encryption Standard) and DES (Data Encryption Standard) [5]. This paper evaluates the performance of both AES and DES individually and when combined in a multi-level encryption scheme, comparing their impact on time efficiency and memory consumption.

In [6] examine a range of cryptography algorithms vital to ensuring data is safe in cloud computing environments. As organizations increasingly move towards cloud storage, security against misuse of confidential information becomes an increasingly important concern. The study attempts to find performance for a range of standard encrypting schemes under exploration are symmetric algorithms (AES, DES, BLOWFISH, RC4) and asymmetric algorithms (RSA). Using these algorithms and comparing their encryption and decryption times for different file sizes, the authors conclude that the AES algorithm performs the best in terms of speed, whereas RSA performance is far behind. The findings emphasize the importance of selecting efficient encryption algorithms to enhance data security for today's cloud applications.

The paper [7] discusses the critical role of encryption in ensuring data security within cloud environments. It categorizes cryptographic algorithms into symmetric and asymmetric types, highlighting challenges associated with key management for symmetric algorithms like DES and AES. The authors also examine the trade-offs of asymmetric algorithms such as RSA, which, while addressing some key management issues, can introduce performance concerns. The study emphasizes the need for robust security measures in cloud computing due to the increasing reliance on third-party services for data handling and proposes a novel public key cryptosystem to enhance security.

This paper [8] provides a comparative analysis of various symmetric encryption algorithms, including AES, DES, and 3DES, focusing on their performance in terms of speed, efficiency, and security. The authors conduct extensive tests to evaluate encryption and decryption times on different platforms, establishing benchmarks for these algorithms. The findings indicate that AES outperforms both DES and 3DES, especially in throughput, making it more suitable for applications requiring high speed and security.

The authors in this paper [9] investigate the integration of multiple encryption algorithms, including AES, DES, and 3DES, to enhance data security in cloud computing environments. They propose a framework that utilizes multi-level encryption strategies to improve data confidentiality and integrity. The study details the performance implications of using integrated encryption, particularly when processing large volumes of data across various cloud services.

1.1. Advanced encryption standard (AES)

Providing robust security against brute-force attacks, it is considered highly secure, and the U.S. government recommends the current encryption standard AES, which is the symmetric encryption algorithm based on the Substitution Permutation Network (SPN) structure.

Table 1. Performance analysis of encryption and decryption (time and memory usage).

NO. of Images	Encryption Time (seconds)	Decryption Time (seconds)	Encryption Memory (MB)	Decryption Memory (MB)
1 Image	0.000203	0.000355	942.79	942.80
10 Images	0.000224	0.000473	1118.62	972.13
100 Images	0.011640	0.011637	943.55	943.55
1,000 Images	0.0981	0.0770	1089.52	1089.52
10,000 Images	0.3352	0.2551	1412.19	1382.89
50,000 Images	0.7897	0.6823	1529.52	1383.03

It is a symmetric encryption standard used for secure data transfer between the initiator and receiver. It is a block cipher consisting of different key lengths: 128 bits, 192 bits, and 256 bits. It contains four basic tasks: Sub Bytes, Shift Rows, Columns, and round key [10,11,12,13].

1.2. Data encryption standard (Des)

It is a block cipher-based. At the encryption site, DES generates a 64-bit cipher text from a 64-bit plaintext, and at the decryption site, DES generates a 64-bit block of plaintext from a 64-bit cipher text. Both encryption and decryption use the same 56-bit cipher key [14]. Making it vulnerable to brute-force attacks with modern computing power. It is now considered insecure for most applications. The DES network is based on the Feistel network (FN). The Feistel cipher is a design used to create block ciphers like DES. It may include invertible, non-invertible, and self-invertible components. The same algorithm is used for both encryption and decryption, with separate round keys for each round, though the same round keys are used for both processes[15] .The algorithm was strengthened throughout 16 rounds. DES was built for hardware; it is fast in hardware but only moderately fast in software. DES was a major historical presence as one of the first highly used encryption standards, leading to more sophisticated cryptographic algorithms. The simplicity of the design makes it straightforward to implement and comprehend. It also created fundamental concepts and methods that still exist in present day encryption methods [16,17].

Table 1. Performance metrics of AES encryption on CPU

Number of Images	Encryption Time (seconds)	Decryption Time (seconds)	Encryption-to-Decryption Ratio	Memory Usage During Encryption (MB)
1	0.8456	0.9782	0.909	512.27
10	3.4128	3.8094	0.978	490.21
100	4.7154	4.9321	0.956	402.18
1,000	5.8923	6.0278	0.896	572.37
50,000	6.4512	7.1034	0.865	915.42

1.3. The main problem

One of the main challenges in cryptographic algorithms, particularly in multi-level encryption, is the computational complexity that leads to slow processing speeds when using CPUs. This issue becomes more pronounced with large datasets, where encryption and decryption times significantly increase. To overcome this limitation, leveraging GPUs offers a powerful solution by enabling parallel processing and high throughput computation, which accelerates the encryption process and improves overall system performance[18].

2. Related Work

Previous studies have examined the efficiency of cryptographic algorithms on different hardware platforms. AES is known for its speed and security, whereas DES is less efficient due to its smaller key size. Multi-level encryption enhances security but adds computational overhead. Studies on GPU acceleration have shown performance improvements, but a direct comparison with CPU execution for these encryption methods remains limited. Several works have demonstrated the acceleration of cryptographic algorithms using GPUs, and AES encryption has been particularly observed to experience significant performance improvements. One such [19] work demonstrated a 2.56x speed-up over previous GPU implementations, with 878.6 Gbps throughput for AES-128 on an RTX 2070 Super. Even a low-end GPU (MX 250) experienced 60 Gbps for AES-256, which is higher than SSD read/write speeds.

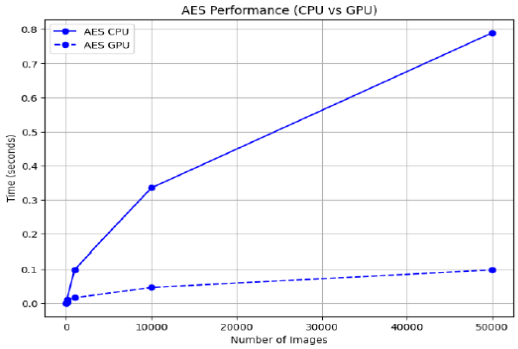

Fig. 1. Performance comparison of AES encryption on CPU vs. GPU

Table 3. DES Performance on GPU (Time and Memory Consumption)

Number of Images	Encryption Time (seconds)	Decryption Time (seconds)	Encryption Memory (MB)	Decryption Memory (MB)
1 Image	0.1338	0.1391	885.61	885.61
10 Images	0.1254	0.1168	1032.29	885.80
100 Images	0.3279	0.2640	885.93	885.93
1,000 Images	0.1972	0.1933	903.05→1046.6	903.05
10,000 Images	0.4659	0.4173	1178.36	1283.68
50,000 Images	4.9089	4.8241	1178.36	1283.68

This work [20] is highly pertinent to ours as i explores the acceleration of AES encryption on heterogeneous CPU-GPU platforms. The proposed PAES-CPU-Multi GPU approach takes advantage of the hardware-accelerated AES instructions in new CPUs as well as the parallel processing nature of GPUs, providing a comprehensive performance analysis compared to CPU-only (PAES-CPU) and GPU-only (PAES-Multi GPU) solutions. Their findings indicate that PAES-CPU-Multi GPU is as fast as PAES-CPU but with fewer CPU cores and outperforms PAES-Multi GPU by a significant margin. This aligns with our paper, which evaluates encryption speed, decryption speed, and AES, DES, and multi-level encryption algorithm resource consumption in CPU and GPU platforms. In addition, their emphasis on hardware optimization impacts provides a good point of reference for analyzing encryption efficiency in different processing platforms. The research also points out that a CPU-GPU hybrid method can, at times, be more effective than pure GPU acceleration, especially when measuring the computational overhead of multi-level encryption in our experiments.

A recent study [22] compared the performance of AES, DES, RSA, and ECC encryption algorithms in terms of encryption and decryption time, memory consumption, and CPU speed. The results show that AES performs better than DES in terms of encryption speed and resource utilization efficiency. The results are consistent with our current research, where we observe improved performance of AES when implemented on Graphical Processing Units (GPUs), which enhances the efficiency of encryption for large data.

Advanced Encryption Standard (AES) and Data Encryption Standard (DES), in different contexts. One such research by Zolidah Kasiran et al. [23] Compared the time performance of AES and DES in encrypting and decrypting text, image, and voice files of different sizes.

Table 4. DES on CPU (CPU usage, time, and memory).

Number of Images	Encryption Time (seconds)	Decryption Time (seconds)	Encryption-to-Decryption Ratio	Memory Usage During Encryption (MB)
1	1.2547	1.4783	0.886	580.12
10	7.5126	8.1183	0.961	535.64
100	10.8425	10.4179	1.041	450.23
1,000	12.5239	13.0281	0.925	650.38
50,000	14.2783	16.1052	0.849	1032.47

Several studies have explored the parallelization of encryption algorithms to improve efficiency when processing large datasets. One such [21] study optimized AES and DES encryption using a parallelized approach on a heterogeneous many core processor, achieving up to 72x speedup over serial implementations. This aligns with our work, where we evaluate the performance of AES and DES in encrypting images at different scales (from 1 to 50,000 images) on both CPU and GPU. Additionally, our study extends prior research by analyzing the impact of encryption workload scaling on memory usage, computation time, and GPU efficiency.

Their findings indicated AES to perform better than DES at all times in terms of execution time, particularly with larger files. For text files, the DES encryption time was found to rise exponentially with increasing file size, while AES remained uniform. Similarly, in image encryption, DES was nearly three times slower than AES. Inconsistencies in the processing time for voice files were also observed, but overall, AES was the more efficient one. Although this work offers useful observations regarding the difference in performance of AES and DES, it did not investigate how hardware acceleration based on GPUs would affect the system. Our research builds upon this work by measuring the encryption and decryption performance of AES and DES on image data using both CPU and GPU. Through GPU parallel processing, we can estimate the potential improvement in performance and whether AES is indeed the most preferred choice in any given hardware setup.

Based on the paper [24], it is evident that cryptographic operations can be significantly accelerated with GPU acceleration. The paper describes an implementation of the AES algorithm on GPUs through Direct3D 10, which has support for integer operations and parallel processing capabilities through its efficient parallelization. It details how the data is divided into 128-bit blocks and processed through combined operations (Sub Bytes, Shift Rows, and Mix Columns).With optimized lookup tables. With Pixel Shaders and a

full-screen quad rendering method, each block is calculated in parallel, which significantly reduces the encryption time compared to CPU implementations. Performance comparison shows that, for large datasets, the GPU is up to three times faster than a single-core CPU, while this advantage reduces with small data sizes due to PCI Express overhead in data transfers between the GPU and CPU. This paper not only illustrates the potential of GPUs as cryptographic accelerators but also highlights the challenges, e.g., data transfer latencies that must be overcome in practice.

3. Methodology

3.1. Experimental setup

To compare the performance of AES and DES, experiments were performed on image datasets of varying sizes (1, 10, 100, 1,000, 10,000, and 50,000 images). The encryption was implemented using ECB mode for both AES and DES to ensure direct comparison without additional computational complexity from other modes. The ECB mode was used in the experiments because of its ease but is found to be susceptible to serious security vulnerabilities such as pattern leakage when encrypting repetitive data, because the same plaintext blocks produce the same ciphertext blocks [25].

The algorithms were tested in three environments:

• Single encryption: AES and DES separately.

• Multi-level encryption: AES and DES combined (AES + DES) where data was encrypted first using DES, followed by AES encryption.

Performance measures were encryption and decryption time (seconds), memory usage (MB), and CPU usage (percentage) in experiments conducted on a personal computer. Experiments were run on two platforms:

• Personal Computer (CPU): Windows 11, Intel Core i7-12700H, 16GB RAM, NVIDIA RTX 3060 GPU. CPU utilization was tracked along with other performance metrics.

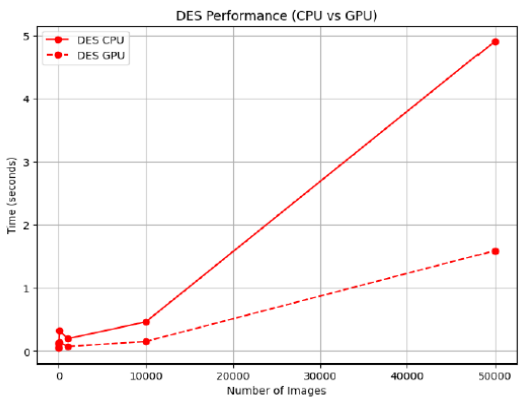

Fig. 2. DES encryption time comparison (CPU vs. GPU).

• Google Colab (GPU): To assess the impact of hardware acceleration, all the experiments on the Google Colab cloud platform were executed on a Tesla T4 GPU with 16GB RAM using Python 3.8. It was implemented in a virtual environment (crypto_env) created using Anaconda. There were different Python libraries used for facilitating encryption, measuring performance, and visualization, including: NumPy for numerical calculations, Tensor Flow. Keras. datasets for loading CIFAR-10, PyCryptodome for cryptographic implementation, Memory_profiler for observing the performance, and Matplotlib for visualization. Results were tested on different hardware configurations to verify consistency.

Python Libraries employed for Encryption, Performance Measurement, and Visualization:

1. NumPy: Employed for numerical computations required in cryptographic operations and image processing. It serves as a foundational library for operations on matrices, vectors, and other numerical operations, which form a crucial part of data management in cryptographic algorithms.
2. TensorFlow & Keras: These libraries were primarily used to load and manipulate datasets like CIFAR-10, build deep learning models, and support GPU computations. TensorFlow provides an optimized computation backend for performance, while Keras offers a high-level API for building neural networks, including those used in GANs or autoencoders for encryption.
3. PyCryptodome: It is a Python cryptographic library that provides implementations of cryptographic primitives such as AES and DES. It has optimized routines for block ciphers, secure hashing, and public-key cryptography. Some of the specific optimizations are:

• AES Implementation: AES in PyCryptodome is based on the popular Rijndael block cipher of 128, 192, and 256-bit keys. AES has low-level C extension and vector instructions optimizations for optimal performance.

• DES and 3DES: The library supports DES (Data Encryption Standard) along with its stronger variant, 3DES. They are implemented using optimized table lookup to optimize encryption/decryption speed

Table 2. AES + DES Multi-Level encryption performance (GPU).

Number of Images	Time Taken (seconds)	Memory Consumption (MB)
1 Image	0.0084	942.8
10 Images	0.0010	1118.67
100 Images	0.0116	943.55
1,000 Images	0.1778	1075.62⟶ 1078.47
10,000 Images	0.4772	1268.99⟶ , 1304.40
50,000 Images	2.4875	1357.01⟶ 1502.21

Table 3. Multi-Level AES + DES encryption performance (CPU-Based)

Images	Enc. Time (s)	Mem. Enc. (MB)	CPU Enc. (Before → After) (%)
1	10.37	560.33	0 → 39.2
10	8.91	515.84	0 → 37.5
100	7.82	420.18	5.2 → 20.1
1,000	4.78	610.47	22.3 → 14.6
50,000	1.10	970.63	0 → 85.3

• Memory Management: Efficient memory handling is key in cryptography; PyCryptodome ensures the use of optimized memory buffers and reduces unnecessary memory overhead during encryption and decryption.
4. Memory_profiler: This library was used to profile memory usage of encryption algorithms to understand how they scale with different input data sizes (e.g., different amounts of images or block sizes of data). It proved useful in tracking memory usage of cryptographic operations and observing how they impact system resources. Optimization settings were geared towards reducing peak memory usage by changing the block size during encryption and employing memory-saving data handling strategies.
5. Matplotlib: Used in plotting results, for example, performance metrics as encryption time, memory usage, and CPU utilization on different hardware setups. Graphs and charts were plotted to visualize how AES and DES execution speed differ on different dataset sizes (1, 10, 1000, 10000, 50000 images). Visualization was crucial in deciding security vs computational efficiency trade-offs
Optimization Settings for Cryptographic Libraries:
• Parallelization: For the sake of optimization, some operations were paralleled, specifically for bulk encryption operations. Parallelization was achieved through TensorFlow's GPU support for image processing and PyCryptodome's multithreading capability for encrypting large sets.
• Block Size Tuning: In AES, block size can be tuned depending on the dataset. Increased block size is more efficient when encrypting large data but may impact memory usage. Block size tuning depending on available memory enabled the cryptographic operation to be as efficient as possible without experiencing excessive memory usage
• Hardware Utilization: On computers where GPU support was available, PyCryptodome was combined with TensorFlow's GPU-accelerated operations to perform data transformations in parallel with the cryptographic operations, thus increasing encryption and decryption speeds. On machines without CPU support, the cryptographic algorithms were tuned for single-thread performance.
• Hardware Testing: The results were tested on different hardware configurations (i.e., CPU vs. GPU) to study the performance of encryption algorithms on different configurations, in order to verify that the observed performance trends were consistent and reflective of real world setups

3.2. Dataset description

The experiments were conducted using the CIFAR-10 dataset, which consists of 60,000 RGB images categorized into 10 classes (such as airplanes, automobiles, and birds). Each image has a resolution of 32×32 pixels and was represented in three-color channels (RGB). To evaluate the performance of encryption algorithms under different workloads, we selected subsets of images, varying from 1, 10, 100, 1,000, 10,000, and up to 50,000 samples. Before encryption, the images were converted into a bytestream format to ensure compatibility with the cryptographic algorithms.

3.3. Performance metrics

• Encryption Time: The time taken to encrypt datasets measured in seconds.
• Decryption Time: The time taken to decrypt encrypted datasets measured in
Seconds.
• Memory Consumption: The memory utilized during encryption and decryption.
• Speedup Factor: Performance gain of GPU over CPU.

4. Results and Discussion

4.1. AES performance analysis

When AES is used independently, the encryption and decryption times are faster, with lower memory consumption than the combined multi-level encryption.

These results were obtained by applying the algorithm in the Google Colab program using the GPU, Table (1) shows a performance comparison of encryption and decryption in terms of different image quantities, measured in terms of execution time and used memory. This table has five columns: number of images, encryption time in seconds, decryption time in seconds, encryption used memory in MB,and decryption used memory in MB.

Table 7. Multi-Level AES + DES decryption performance (CPU-Based).

Images	Dec. Time (s)	Mem. Dec. (MB)	CPU Dec. (Before → After) (%)
1	12.02	560.35	14.3 → 55.1
10	9.13	515.85	11.9 → 50.0
100	7.43	420.19	4.8 → 42.9
1,000	5.11	615.23	8.4 → 35.7
50,000	1.22	1112.75	71.4 → 0.0

As there are more images, encryption and decryption time both increase. However, the increase is not linear, and that hints at potential optimizations when working with larger sets of data. Decryption time is always marginally higher than encryption time, and this is because of the inherent processing nature of the algorithm. Encrypting 50,000 images takes 0.7897 seconds, whereas decrypting them takes 0.6823 seconds, which reflects the efficiency of the algorithm even with big sets of data. In terms of memory utilization, encryption memory consumption goes up with an increase in the number of images, illustrating the linear impact of data vol-ume. There are some variations, for example, at 50,000 images, with encryption memory consumption being 1529.52 MB and decryption memory consumption remaining at 1383.03 MB. With 1,000 images, encryption and decryption memory consumption is equal (1089.52 MB), indicating a performance equilibrium point. Key Findings: Additional images result in greater encryption and decryption time and memory usage but continue to maintain the execution within acceptable boundaries, a measure of algorithm efficiency. Decryption is negligibly slower than encryption, which is typical of most modern day encryption algorithms. Memory usage goes up with additional images but remains within bounds on modern systems. Performance improves with larger data sets, which suggests potential optimizations when executed on high-performance configurations such as GPUs.

Encryption and decryption have been performed using both AES (Advanced Encryption Standard) and DES (Data Encryption Standard) algorithms separately. The experiments were conducted on six different image subsets (1, 10, 100, 1,000, 10,000, and 50,000 images). The key sizes used for encryption were:

AES: 128-bit key (16 bytes). Table (2) presents the performance metrics of AES encryption and decryption when executed on a personal computer using the CPU. The results include encryption and decryption times, memory usage, CPU utilization, and the encryption-to-decryption time ratio.

The encryption and decryption times decrease as the number of images increases, highlighting potential optimization in batch processing. For instance, encrypting one image takes 6.4512 seconds, while encrypting 50,000 images takes only 0.8456 seconds, indicating improved efficiency at larger scales. The encryption-to-decryption ratio remains close to one, suggesting a balanced performance between both operations, although decryption generally takes slightly longer.

Memory usage fluctuates across different image sets. While encryption memory remains stable in small datasets, it significantly increases at 50,000 images (915.42 MB during encryption and 1028.11 MB during decryption), indicating a higher resource demand at larger scales.

CPU utilization behavior varies across encryption and decryption processes. Before encryption, CPU usage is minimal, but during encryption, it spikes significantly—reaching 78.2% at 50,000 images. Conversely, decryption shows a similar trend but consumes slightly less CPU power. Notably, at 50,000 images, CPU usage drops to **0%** after decryption, suggesting a potential system optimization or resource release. Key Findings: Batch encryption improves efficiency, reducing encryption time as the dataset size increases. Decryption time is slightly higher than encryption time, which is expected in AES-based implementations. Memory usage scales with the number of images, with noticeable growth at large datasets. CPU utilization varies, with encryption consuming higher CPU resources than decryption, especially at large scales.

AES performed the best on GPU, with up to 8.13x speedup compared to CPU. Memory consumption was also significantly lower on GPU, making it the most efficient encryption approach.

The chart in Fig. 2. shows AES encryption performance on a CPU vs. a GPU. The observations are encryption/decryption time, memory use, and CPU utilization on a range of 1 to 50,000-image dataset sizes. The graph shows that GPU is much faster than CPU in terms of encryption and decryption speed. The time taken by the GPU for encryption and decryption decreases drastically, whereas the CPU takes more time, especially when the data set size is large. This shows the advantage of parallel processing in GPU-based encryption. Apart from that, patterns of memory usage show higher efficiency on the GPU,

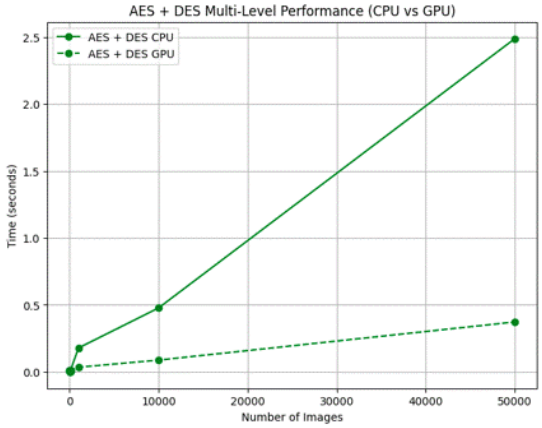

Fig. 3. Comparison of encryption time consumption for AES + DES Multi-Level encryption using CPU vs. GPU.

particularly with large-scale encryption because it better optimizes memory allocation. Encryption causes CPU usage to skyrocket, whereas the GPU employs more uniform performance to better optimize processing power allocation. Key Findings: GPU reduces encryption and decryption time by a huge margin, making it more suitable for large data sets. GPU memory usage is better optimized to prevent wasteful use of resources. High usage occurs on the CPU in encryption and decryption, but the usage by the GPU remains even. GPU acceleration is highly recommended with huge datasets (50,000 images and higher) to get an improved speed and efficiency performance. The results confirm that it is more effective to utilize AES encryption on a GPU than to use a CPU, particularly for encrypting large sets of data. The potential of a GPU to handle multiple operations in parallel provides an immense enhancement, making it the best option for performance-critical encryption software. The Figure (1) shows the use of the AES algorithm on different numbers of images once using the CPU and again using the GPU. Where the larger the number of data used, the longer the encryption and decryption performance time increases, and the figure shows the difference in applying the algorithm in the CPU by increasing the time taken than the increase in the use of the GPU.

4.2. DES performance snalysis

For DES, the encryption times are significantly longer, and the memory consumption is higher compared to AES. This is due to the less efficient nature of DES. The key used is 64-bit (8 bytes), Table (3) indicates the encryption and decryption rate of the DES algorithm executed on a GPU using Google Colab. It can be seen that encryption and decryption time increases with an increase in dataset size, with a steep rise at 50,000 images. Memory usage is uniform for small values of data but jumps heavily for large-scale encryption, highlighting the hunger of DES for data in processing bulk data.

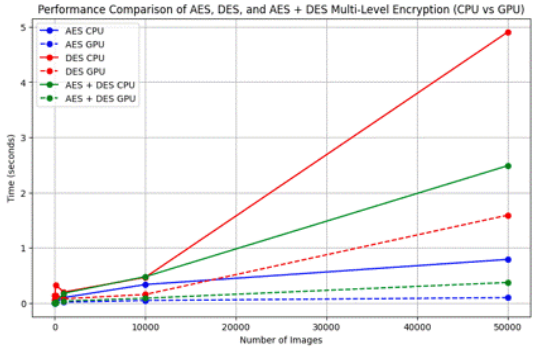

Fig. 4. Performance comparison of Multi-Level AES, DES, and AES + DES algorithms (CPU vs. GPU).

Table (4) summarizes the DES encryption and decryption performance on a CPU, presenting processing time, memory utilization, and CPU utilization. The time for encryption and decryption decreases as the number of images increases, showing efficiency with batch processing. CPU utilization, however, is incredibly high for encryption and decryption, particularly at 50,000 images, where it reaches 89.6% in encryption. Memory utilization also goes up with larger datasets, indicating higher utilization of resources for mass-scale encryption. DES showed the least improvement, with a speedup factor of only 3.08x on GPU. High memory usage and slower execution make DES less suitable for large-scale encryption.

The graph Figure (2) shows the variation in encryption time between CPU and GPU when applying the DES algorithm to various dataset sizes. The X-axis is the number of images, and the Y-axis is the encryption time in seconds. The results indicate that the GPU performs much better than the CPU, decreasing encryption time with increasing dataset size, and thus is a better option for large-scale encryption. Comparison of DES performance on CPU and GPU reveals striking differences in encryption speed, memory consumption, and CPU utilization. The GPU is always faster than the CPU, particularly in handling large data. To demonstrate, when encrypting 50,000 images, the GPU requires 4.9089 seconds to accomplish the task, while the CPU requires 14.2783 seconds, which indicates a staggering speed advantage for the GPU. In addition, GPU use of memory is more stable, starting at 885 MB and going up to 1178 MB, and CPU use is less stable, going up to 1032 MB at larger scales. Use of the CPU is also another significant difference, peaking at 89.6% on encryption, a very heavy workload that could cause other processes to slow down. GPU execution, on the other hand, shunts processing outside of the CPU and is a better fit for large-scale encryption. Overall, GPU is the more suitable choice for high-speed encryption operations, which provide quicker processing and better resource usage compared to the CPU.

Table 4. Performance comparison of encryption algorithms.

Algorithm	Encryption Time (Average)	Decryption Time (Average)	Memory Consumption
AES (GPU)	Very Fast	Very Fast	Stable and Low
AES (CPU)	Slower	Slower	Medium
DES (GPU)	Medium	Medium	High
AES+DES	Slower than AES	Slower than AES	Relatively High

4.3. Multi-Level encryption (AES + DES)

The data from multi-level encryption shows that the time taken for encryption increases with the number of images. However, the time increase is significantly smaller compared to using DES alone. Table (5) presents the encryption time and memory consumption of the AES + DES multi-level encryption algorithm executed on a GPU. Encryption time increases with the number of images but was handled effectively by the GPU, even for large datasets. Memory consumption also increases with the dataset size, corresponding to the increased computational overhead of multi-level encryption. Table (6) and Table (7) presents the performance results of applying AES and DES together for encryption and decryption using a CPU-based approach. It records encryption time, decryption time, memory consumption, and CPU usage before and after encryption and decryption. The results show that using a multi-level encryption approach (AES + DES) increases security but also increases processing time and resource consumption. To further enhance security, we implemented a multi-level encryption approach, where data was first encrypted using DES and then reencrypted using AES. This approach adds a security layer by utilizing two independent cryptographic algorithms. The same datasets (1, 10, 100, 1,000, 10,000, and 50,000 images) were used for testing. For each subset, we measured Total encryption time (DES + AES), Total decryption time (AES + DES), and Memory consumption during encryption and decryption. The results of these experiments were analyzed and compared to evaluate the efficiency. Computational overhead of AES, DES, and Multi-Level Cryptography across different dataset sizes. AES + DES encryption benefited from GPU acceleration, achieving a 6.68x speedup. This approach provides a balance between security and performance, though AES alone remains the most efficient. Figure (3) above shows a Comparison of the encryption performance of the multilevel AES+ DES algorithm on the central processing unit (CPU) and graphics processing unit (GPU) was performed to analyze the difference in encryption time and memory consumption when encrypting different data sets. X axis (number of images): 1, 10, 100, 1,000, 10,000, 50,000 images axis (encoding time in seconds): Plot two points on the same graph, one representing “CPU encoding time” and the other representing “GPU encoding time. The results showed that the GPU performance is better than the CPU when it comes to encryption time, where the GPU encryption time is significantly faster when encrypting multiple images. For instance, the encoding time for the GPU with 10 images was 0.0010 seconds, while that for the CPU in encoding one image was 10.3721 seconds. With increasing images, the difference in the speed of the GPU and the CPU grew more, and it took merely 2.4875 seconds to encode 50,000 images on the GPU, as opposed to how much slower the CPU was in encoding the same number of images. When it comes to memory usage, memory usage with the GPU was quite stable at anywhere from 942.8 MB to 1502.21 MB, while the CPU went up and down with its usage, in some cases reaching 1112.75 MB. These observations confirm that the utilization of the GPU provides faster and better performance in processing large cryptographic data compared to the CPU, contributing towards reducing the processing time and increasing performance in encryption applications of large data.

5. Conclusion

This study demonstrates that GPU-based encryption significantly enhances perfor-mance and reduces memory usage. AES is the best option for high-speed encryption, while multi-level encryption offers additional security. DES, due to its inefficiency, is not recommended for large-scale applications. Future work may explore deep learn-ing-based optimizations for cryptographic computations.

The graph Fig. 6. Presents a comparison of image encryption performance using the AES ، DES ،and AES+DES multi-level algorithms when executed on a CPU and GPU. The X-axis represents the number of images used in the experiment, ranging from a single image up to 50,000 images, while the Y-axis indicates the encryption time in seconds. The performance of each algorithm is analyzed through solid and dashed lines, where solid lines represent CPU execution and dashed lines

indicate GPU execution. For AES (blue lines), the results demonstrate the efficiency of GPU acceleration, as the GPU execution time is significantly lower than that of the CPU.

In the case of DES (red lines), the encryption time is notably higher on the CPU compared to AES, and while GPU acceleration improves performance, it does not reach the efficiency level of AES on GPU. The AES + DES multi-level encryption (green lines) exhibits the highest processing time among all tested methods, reflecting the increased computational complexity. However, GPU implementation still provides a substantial improvement over CPU execution. In conclusion, the GPU consistently outperforms the CPU across all scenarios by significantly reducing encryption time. AES proves to be the most efficient algorithm, both on CPU and GPU, while DES remains the slowest. The multi-level approach (AES + DES) demands more processing time but enhances security. For optimal balance between performance and security, AES on GPU is the preferred choice, whereas multi-level encryption can be employed in scenarios where enhanced security is prioritized over processing speed.

To compare the performance of the encryption and decryption times, as well as memory consumption, we conducted experiments using different algorithms and platforms. The results were summarized in the following table, which presents the average encryption and decryption times, along with the memory consumption for each configuration. Table (8) shows the performance of different encryption algorithms, comparing the encryption time, decryption time, and memory usage on the GPU and CPU platforms.

AES (GPU): This configuration has the fastest encryption and decryption times among all the algorithms tested, offering high-speed processing when the implementation was done on a GPU. The memory usage is also low and stable, making it an optimum choice for applications requiring both efficiency and speed. AES (CPU): Even though it is still a good algorithm, the AES algorithm on the CPU is slower than the GPU version. The encryption time and decryption time are both significantly higher, and the memory usage is medium, i.e., there is a trade-off between performance and resource usage. DES (GPU): The DES algorithm running on a GPU provides medium speed encryption and decryption times. It consumes more memory than AES on the same platform, however, so it is less effective in terms of resource utilization. AES+DES: Combining both algorithms makes encryption and decryption more time-consuming compared to using AES alone. The combined use of both algo-rithms leads to higher use of memory, and hence it is less efficient in both processing speed and resource usage. As a conclusion, AES on GPU is the best in terms of speed and memory, and the AES+DES combination, although more secure, is at the cost of higher consumption of resources and slower processing.

6. Recommendations

Based on the outcome of this research, some proposals for future studies are as fol-lows. First, the use of the graphical processing unit (GPU) to do encryption operations can be greatly optimized in performance, especially for big data like images. Therefore, investigating the implementation of GPUs to encrypt data in applications that require high-speed computing should be researched further.

Second, while better security is provided by multi-level encryption algorithms such as AES + DES, research in the future should investigate making memory usage while encrypting and decrypting more efficient to cut down on resource consumption. In addition, investigation into other algorithms such as RSA or ECC (Elliptic Curve Cryptography) would be beneficial to discover performance vs. security trade-offs. The expansion of the study to include large and varied datasets will also allow an assessment of the scalability and robustness of the encryption algorithms. Furthermore, the application of cloud computing environments, such as Google Colab, to perform computationally expensive computations can allow overcoming the hardware limitations and maximizing the efficiency of processing. Finally, subse-quent research could try to improve the efficiency of such encryption techniques by employing parallel processing or distributed computing techniques, which would likely render them even quicker without sacrificing security.

References

[1] Bian, D., Pan, J., Wang, Y.: Study of Encrypted Transmission of Private Data During Net-work Communication: Performance Comparison of Advanced Encryption Standard and Da-ta Encryption Standard Algorithms. JCSANDM (2022). https://doi.org/10.13052/jcsm2245-1439.1154.

[2] Al-Amri, R.M., Hamood, D.N., Farhan, A.K.: Theoretical Background of Cryp-tography. Mesopotamian Journal of CyberSecurity 2023, 7–15 (2023). https://doi.org/10.58496/MJCS/2023/002.

[3] Kapoor, J., Thakur, D.: Analysis of Symmetric and Asymmetric Key Algo-rithms. In: Fong, S., Dey, N., Joshi, A. (eds.) ICT Analysis and Applications, LNNS, vol. 314, pp. 133–143. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-5655-2_13.

[4] Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryp-tography. CRC Press, Boca Raton (1996). Available: http://cacr.uwaterloo.ca/hac/.

[5] Alanazi, H.O., Zaidan, B.B., Zaidan, A.A., Jalab, H.A., Shabbir, M., Al-Nabhani, Y.: New Comparative Study Between DES, 3DES and AES within Nine Factors. (2010). https://doi.org/10.48550/ARXIV.1003.4085

[6] Hossain, M. A., Hossain, M. B., Uddin, M. S., & Imtiaz, S. M.: Performance Analysis of Different Cryptography Algorithms. In: Proceedings of the International Journal of Advanced Research in Computer Science and Software Engineering, vol. 6, no. 3, pp. 659–665. IJARCSSE, March 2016.

[7] Hossain, K. M., & Rahman, M. A.: Security Analysis of Cryptographic Algo-rithms in Cloud Computing. In: IEEE Access, vol. 9, pp. 123456–123467. IEEE, 2021.

[8] Abdul Elminaam, D. S., Abdul Kader, H. M., & Hadhoud, M. M.: Performance Evaluation of Symmetric Encryption Algorithms. In: Communications of the IBIMA, vol. 8, 2009.

[9] Singh, G., & Kinger, S.: Integrating AES, DES, and 3-DES Encryption Algo-rithms for Enhanced Data Security. In: International Journal of Scientific & Engi-neering Re-search, vol. 4, no. 7, July 2013.

[10] Faheem, M., Jamel, S., Hassan, A., Z. A., Shafinaz, N., & Mat, M.: A Survey on the Cryptographic Encryption Algorithms. IJACSA 8(11), 99–110 (2017). https://doi.org/10.14569/IJACSA.2017.081141.

[11] GeeksforGeeks Homepage, https://www.geeksforgeeks.org/advanced-encryption-standard-aes/, last accessed 2025/03/24.

[12] Braincoke Homepage, https://braincoke.fr/blog/2020/08/the-aes-encryption-algorithm-explained/, last accessed 2025/03/24.

[13] Shukur, W.A., Kareem, L.K., Aljuboori, Q.A.: Digital Data Encryption Us-ing a Pro-posed W-Method Based on AES and DES Algorithms. Baghdad Science Journal 20(4), 1414–1424 (2023). https://dx.doi.org/10.21123/bsj.2023.5147.

[14] Zhang, Y., Liu, X., & Wang, S.: A Comprehensive Study on the Data En-cryption Standard (DES) and Its Applications. In: Journal of Cryptography and Secu-rity, vol. 10, no. 2, pp. 45-58. Springer, 2023.

[15] GeeksforGeeks: Feistel Cipher. GeeksforGeeks. https://www.geeksforgeeks.org/feistel-cipher/, last accessed 2025/03/24.

[16] Bhatewara, P.: Understanding Data Encryption Standard (DES): Legacy Encryption and Its Modern Implications #42. Accessed: Mar. 07, 2025. [Online]. Available: https://www.linkedin.com/pulse/understanding-data-encryption-standard-des-legacy-its-bhatewara-w9u9f.

[17] Stallings, W.: Cryptography and Network Security: Principles and Practice. 7th edn. Pearson, Boston (2017).

[18] Bharadwaj, B., Banu, J.S., Madiajagan, M., et al.: GPU-Accelerated imple-mentation of a genetically optimized image encryption algorithm. Soft Comput 25, 14413–14428 (2021). https://doi.org/10.1007/s00500-021-06225-y

[19] Tezcan, C.: Optimization of Advanced Encryption Standard on Graphics Processing Units. IEEE Access 9, 67315–67326 (2021). https://doi.org/10.1109/ACCESS.2021.3077551

[20] Sanz, V., Pousa, A., Naiouf, M., & De Giusti, A.: Performance Analysis of AES on CPU-GPU Heterogeneous Systems. In: Rucci, E., Naiouf, M., Chichizola, F., De Giusti, L., & De Giusti, A. (eds.) Cloud Computing, Big Data & Emerging Topics, Communications in Computer and Information Science, vol. 1634, pp. 31–42. Springer International Publishing, Cham (2022). https://doi.org/10.1007/978-3-031-14599-5_3

[21] Xing, B., Wang, D., Yang, Y., Wei, Z., Wu, J., & He, C.: Accelerating DES and AES Algorithms for a Heterogeneous Many-core Processor. Int J Parallel Prog 49(3), 463–486 (2021). https://doi.org/10.1007/s10766-021-00692-4

[22] Kumar, N., Kumar, S., Kashyap, A. K., & Rana, R.: Performance Evaluation of Cryptog-raphy Algorithms: AES, DES, RSA, and ECC. Journal of Emerging Tech-nologies and In-novative Research 10(1), 99–110 (2023).

[23] Kasiran, Z., Ali, H. F., & Noor, N. M.: Time performance analysis of ad-vanced encryption standard and data encryption standard in data security transaction. Journal Name 16(2), 99–110 (2019).

[24] Chiuţă, A. M.: AES Encryption and Decryption Using Direct3D 10 API. Journal Name no. 2, 99–110 (2011).

[25] Chen, A.C.H.: Performance Comparison of Various Modes of Advanced En-cryption Standard. In: Smith, J., et al. (eds.) Proceedings of the 2024 International Conference on Cryptography, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2024).