ISSN: 2182-2069 / E-ISSN: 2182-2077

Performance Evaluation of Contemporary Block Ciphers for IoT Applications

Pericherla Satya Suryateja[1*], and Dr. Kasukurthi Venkata Rao[2]

[1*] Research Scholar, Department of CS & SE, Andhra University College of Engineering, Andhra University, India. suryateja.pericherla@gmail.com, https://orcid.org/0000-0001-8271-6683

[2] Professor, Department of CS & SE, Andhra University College of Engineering, Andhra University, India. professor_venkat@yahoo.com, https://orcid.org/0009-0005-9876-3878

Received: October 30, 2024; Revised: November 27, 2024; Accepted: January 04, 2025; Published: February 28, 2025

Abstract

The Internet of Things (IoT) is paving its way into every aspect of human life. It makes the conventional processes smart and enables them to be completed quickly. Data is of paramount importance in IoT as it is critical for business progression. The security of such data is also crucial to prevent data breaches and loss of privacy. One of the common way to implement data security is Cryptography. Performance evaluation of ciphers like DES, 3DES, AES, Blowfish and Twofish is carried out. The aim of this paper is to evaluate the performance of existing popular ciphers rather than selecting Light Weight Cryptography (LWC) algorithms which are yet to be properly evaluated for their security. A powerful desktop PC and a Raspberry Pi with limited resources that can be considered as a high-end IoT device are selected for evaluating the selected ciphers. From the results, it is observed that Twofish performs better on IoT devices for smaller amounts of data, and it is also memory efficient than other evaluated ciphers. AES and Twofish performed better than other algorithms for various data input sizes. The results of this evaluation will be helpful for IoT architects in making decisions over the required cipher for securing the data in an IoT application.

Keywords: Performance Evaluation of Ciphers, Comparison of Cryptographic Algorithms, Performance Evaluation of Block Ciphers, Data Security in IoT, Ciphers Evaluation on IoT Devices.

1 Introduction

IoT is a technology aggregating data from various things onto a single platform using existing Internet infrastructure. IoT helps in the autonomous exchange of information between uniquely identifiable real-world embedded devices, which are connected by technologies like Radio-Frequency Identification (RFID) and Wireless Sensor Networks (WSNs) (Sreevidya & Supriya, 2024). The data is further processed for decision-making, based on which autonomic actions are performed (Farooq et al., 2015). Different applications of IoT are smart traffic systems, smart environments, smart homes, smart hospitals, smart agriculture, smart parking, augmented maps, smart logistics, smart water supply, smart retail, and supply-chain management, to name a few (Shah & Yaqoob, 2016; Hassija et al., 2019).

Even though IoT improves the quality of life with the help of smart services, it comes with different security-related issues that have detrimental effects on the service. Some IoT-related issues include

Journal of Internet Services and Information Security (JISIS), volume: 15, number: 1 (February), pp. 16-31.
DOI: *10.58346/JISIS.2025.I1.002*

*Corresponding author: Research Scholar, Department of CS & SE, Andhra University College of Engineering, Andhra University, India.

unauthorized access by exploiting RFID vulnerabilities and security breaches in sensor nodes using attacks like jamming, Sybil, tampering, flooding, and other attacks (Farooq et al., 2015; Farhan et al., 2017). IoT also has specific security issues related to privacy, authentication, management, and information storage, to name a few (Aswathy, 2024).). Different types of attacks on IoT layers like the sensing layer, network layer, middleware layer, gateway, and application layer are provided (Hassija et al., 2019; Shah & Sengupta, 2020). Researchers are working more on mitigating the authentication and trust management attacks in IoT (Hassan, 2019).

One of the classic problems related to network security is preventing unauthorized access or providing confidentiality. The general mechanism used to provide confidentiality is encryption by using cryptography. IoT employs a wide range of various heterogeneous devices. In IoT, most devices contain low resources and are often constrained in different ways. So, conventional cryptographic primitives are not suitable for such resource-constrained devices. Hence, lightweight cryptography can efficiently overcome this problem (Hassan, 2019; Thakor et al., 2021).

Various heterogeneous devices used in IoT can be classified into low-end devices, middle-end devices, and high-end IoT devices (Ojo et al., 2018). Low-end devices are too constrained regarding resources like RAM and processing power. They can't run traditional operating systems like Windows or Linux. Examples of low-end devices are OpenMote-B, Zolertia Z1, etc. Middle-end IoT devices have more RAM and processing power than low-end devices and can perform tasks like running low-level computer vision algorithms. They can also have more than one communication technology. High-end IoT devices have larger RAM, CPU, and storage capacities. They can run heavy Machine Learning (ML) algorithms. Due to their higher level of resources, they can often be used as gateways.

As lightweight cryptography is the new norm for providing security in IoT devices, researchers are developing several lightweight algorithms (Purnama et al., 2024). Hence, there is a need to standardize these algorithms and choose some candidate algorithms for further development and testing against attacks (Ofoghi, 2015). Benchmarking is a process that allows the developers of the new lightweight algorithms to compare their algorithms against existing best-performing algorithms. One existing algorithm offering the best performance is the AES GCM implementation (Renner et al., 2019).

With the rise in the adoption of IoT across all domains, the security of data being communicated between IoT devices is paramount (Kavitha, 2024). The mechanism that is generally used to provide data confidentiality is encryption. Due to the resource-constrained nature of IoT devices, contemporary cryptographic algorithms might not be suitable for securing them (Suryateja & Rao, 2024). This research explores some of the famous contemporary algorithms like DES, TDES, AES, Blowfish, and Twofish and their performance for encrypting/decrypting different text sizes. To the best of our knowledge, this is the primary work that compares the performance of these cryptographic algorithms both on a PC and RPi which is a high-end IoT device.

The general architecture of any IoT system or application involves a sensing layer consisting of sensors and actuators, a network layer consisting of an IoT controller like an RPi or a gateway, and the application layer or cloud layer consisting of data processing components (Papadopoulos & Christodoulou, 2024). The network between the IoT controller and the sensors or actuators is typically considered a Local Area Network (LAN), and the network between the IoT controller and the cloud is considered a Wide Area Network (WAN). We assume that the communications within the LAN are secure as the owner of the IoT system often has complete control over it and can safeguard the data transmissions within the LAN. Various security attacks can be performed on the data transmitted over WAN, which the owner has no control over. So, it is essential to provide security and enforce privacy of the data transmitted over the WAN. The favored mechanism for securing data transmission is

encryption. Our findings help IoT practitioners choose appropriate ciphers to secure data transmission between an IoT controller like RPi and the cloud over a WAN.

The content in the rest of the manuscript is organized as follows: The related work section describes the performance evaluation work carried out by researchers already in the existing literature and how this work is different from theirs. The methods & experimental design section is about the methods and experimental design. It thoroughly describes the various steps in the experimental design and its execution. The results and discussion section discusses the results obtained from evaluating the cryptographic algorithms. Finally, the conclusions and future scope section provides the conclusions arrived at and the future scope for extending this work.

2 Related Work

In this section we present a survey of the previous works related to performance analysis of contemporary and lightweight ciphers. Authors in (Padmavathi & Kumari, 2013) compared DES, AES, and RSA in combination with the Least Significant Bit (LSB) substitution technique and measured their performance. The results concluded that AES performance is better than DES and RSA. Authors in (El-Haii et al., 2018) performed a benchmark of block ciphers like DES, 3DES, and AES on a Raspberry Pi 3 and compared them with existing Arduino benchmarks. AES offered the best performance among the block ciphers. Authors in (El-hajj et al., 2023) benchmarked different lightweight algorithms that were submitted to NIST. Analysis of these algorithms was done on Raspberry Pi and Arduino Uno boards.

Authors in (Fotovvat et al., 2020) compared the performance of 32 LWC algorithms on IoT platforms like Raspberry Pi 3, Raspberry Pi Zero W, and iMX233. They concluded that permutation-based ciphers took longer time to execute. Authors in (Abd Elminaam et al., 2010) evaluated most common block ciphers like AES, DES, 3DES, RC2, Blowfish, and RC6. They used a laptop with a 2.4 GHz CPU for experimentation and data collection. Blowfish was identified as the best-performing algorithm. Authors in (Umaparvathi & Varughese, 2010) presented a comparison of symmetric block ciphers like AES, DES, 3DES, and Blowfish regarding power consumption, throughput and other metrics. They used a laptop with an Intel Pentium Core 2 Duo 2.00 GHz processor for conducting experiments. Based on the results, AES performed better than other ciphers.

Authors in (Rizvi et al., 2011) evaluated the performance of AES and Twofish algorithms. They measured the encryption speed for different data types and throughput with varying sizes of RAM. They used an Intel Pentium Dual Core 2.50GHz CPU with 2GB and 4GB RAM for experimentation. The result of the experimentation was interesting. Initially, AES offered better performance than Twofish for less amount of RAM. But, as the RAM size increased, Twofish became faster than AES. Authors in (Haque et al., 2018) evaluated ciphers like AES, RC4, Blowfish, CAST, 3DES, and Twofish for performance on different parameters like key size, data blocks, and encryption/decryption speed. The input was run through 100 times, and the average encryption speed was considered for consistency. They concluded that Twofish and RC4 outperform the other algorithms.

Authors in (Al Tamimi, 2006) compared the performance of four of the most common encryption algorithms, namely, DES, 3DES, Blowfish and AES. Experiments were conducted using a AMD 64-bit processor with 1GB of RAM. The simulation results showed that Blowfish performs better than other encryption algorithms. Authors in (Hossain et al., 2016) compared encryption algorithms like AES, DES, Blowfish, DES, RC4, and RSA for different file sizes in the local system. Evaluation was

conducted on an Intel Core i5 (2.40 GHz) fourth-generation processor with 4GB of RAM with 1 TB-HDD. From the analysis result, the AES algorithm is better than the DES and RSA algorithms.

Authors in (Centeno et al., 2018) compared the performance of four encryption algorithms, AES, Twofish, RSA and ElGamal, on a smartwatch. The results show that applying a specific encryption algorithm has no statistically significant negative impact on the smartwatch's performance. Also, the results concluded that AES offers better performance on a smartwatch platform for encryption. Authors in (Verma & Singh, 2012) compared the performance of AES, RC6 and Twofish algorithms based on execution time and resource utilization. The performance of these algorithms was measured on a 3GHz Pentium 4 processor with 1GB of RAM. Results conclude that RC6 is the best choice when high throughput is needed, and AES is the best choice when RAM size is considered.

Authors in (Sharif & Mansoor, 2010) compared encryption algorithms like Blowfish, CAST-5, IDEA, RC2, RC5, and Serpent based on different key and data sizes. This study used a desktop computer with a 3.06 GHz processor. It was concluded that RC4 performs better than other block cipher algorithms based on the results. Authors in (Vyakaranal & Kengond, 2018) conducted a comparison of symmetric cryptographic algorithms like DES, 3DES, AES, and Blowfish on different factors like encryption time, decryption time, memory usage, etc. A computer with an Intel i3 processor and 4GB RAM was used for conducting experiments. They concluded that AES offers better overall performance.

Authors in (Elminaam et al., 2009) evaluated six of the most common encryption algorithms AES, DES, 3DES, RC2, Blowfish and RC6. The performance data was collected using a laptop with an Intel Pentium IV 2.4 GHz CPU. The result concludes that Blowfish, followed by RC6, performed better than other algorithms. Authors in (Kansal & Mittal, 2014) analyzed symmetric encryption algorithms like AES, DES, and 3DES on an Intel Core i7 processor. The performance of AES was far better than other algorithms. It was observed that AES consumed more RAM than DES. Authors in (Saraiva et al., 2019) presented the performance evaluation of AES, RC6, Twofish, SPECK128, LEA, and ChaCha20-Poly1305 algorithms on various IoT devices. Hardware-accelerated AES was more efficient than every other algorithm. The authenticated stream cipher ChaCha20-Poly1305 performed even better than the block ciphers, consuming less battery while being faster.

Table 1 compares the work done in this paper with previous works. The table contains information about the different cryptographic algorithms evaluated (only block ciphers were mentioned), the hardware used for algorithm analysis, software (like IDEs and libraries), programming languages used, and whether algorithms are evaluated on PCs and IoT devices.

Table 1: Summary of Literature Review on Performance Analysis of Cryptographic Algorithms

Ref.	**Algorithms evaluated**	**Hardware used**	**Software used**	**Evaluation on both PC and IoT devices?**
(Padmavathi & Kumari, 2013)	DES, AES and RSA	NA	Visual studio packages	No
(El-Haii et al., 2018)	DES, 3DES, AES, and others	RPi 3B with 1GB RAM	Raspbian OS and OpenSSL crypto library	No
(El-hajj et al., 2023)	Various NIST lightweight ciphers	Arduino Uno and RPi	C language	No
(Fotovvat et al., 2020)	32 lightweight ciphers	RPi 3, RPi Zero W and iMX233	Linux OS and C language	No

(Abd Elminaam et al., 2010)	AES, DES, 3DES, RC2, Blowfish, RC6	Intel Pentium IV 2.4 GHz CPU	NA	No
(Umaparvathi & Varughese, 2010)	AES, DES, 3DES and Blowfish	Intel Pentium Core 2 DUO 2.0 GHz CPU	Java language	No
(Rizvi et al., 2011)	AES and Twofish	Intel Pentium Dual Core 2.50GHz CPU with 4GB RAM	Win XP OS and C# language	No
(Haque et al., 2018)	AES, RC4, Blowfish, CAST, 3DES, and Twofish	NA	MATLAB and Python language with PyCrypto 1 and Chilkat 2 packages	No
(Al Tamimi, 2006)	AES, DES, 3DES, and Blowfish	3500+ AMD 64-bit CPU with 1GB of RAM	C# language	No
(Hossain et al., 2016)	AES, DES, 3DES, Blowfish, and RC4	Intel Core i5 2.4 GHz CPU with 4GB RAM	Win 8.1 Pro OS, Java language, and MATLAB	No
(Centeno et al., 2018)	AES and Twofish	Samsung Gear S3 smartwatch	NA	No
(Verma & Singh, 2012)	AES, RC6, and Twofish	Intel Pentium IV 3 GHz CPUwith 1 GB RAM	Win XP OS, C# language	No
(Sharif & Mansoor, 2010)	IDEA, Blowfish, RC2, RC5, Serpent and CAST-5	3.06 GHz CPU	Unix, Java language with Bouncy Castle crypto library	No
(Vyakaranal & Kengond, 2018)	AES, DES, 3DES and Blowfish	Intel Core i3 CPU with 4GB RAM	Java language	No
(Elminaam et al., 2009)	AES, DES, 3DES, RC2, Blowfish, and RC6	Intel Pentium IV 2.4 GHz CPU	C# language	No
(Kansal & Mittal, 2014)	AES, DES, and 3DES	Intel Core i7 CPU	C language	No
(Saraiva et al., 2019)	AES, RC6, Twofish, and others	Samsung Galaxy Core Prime and Xiomi Redmi Note 3	Android OS, Java & C++ languages, Crypto++ 8.2 library	No
This Paper	DES, 3DES, AES, Blowfish, and Twofish	Intel Core i5 2.40GHz CPU with 16 GB RAM and RPi with 1GB RAM	Win 10 Pro and Yocto (Linux) OS, Java language	Yes

3 Methods and Experimental Design

Cryptographic algorithms like DES, TDES, AES, Blowfish, and Twofish are evaluated by running each algorithm on the input data multiple times and measuring the metrics. These algorithms are evaluated on a high-end desktop PC and a near-low-end RPi.

Evaluated Algorithms

The DES is a standard National Institute of Standard and Technology (NIST) proposed for securing data communication over a network. The actual algorithm is called the Data Encryption Algorithm (DEA). DES is a symmetric block cipher that processes the plaintext in 64-bit blocks. The effective key length in DES is 56 bits, and the message is processed through 16 rounds of operations. Although at the time of the proposal, the security of DES was quite adequate, over the years, brute force on the DES key space became much more accessible. Hence, DES was extended to Triple DES (TDES) by using DES three times with three different keys. NIST later replaced DES with TDES as a secure communication standard. The effective key size of TDES is 168 bits, which is quite adequate to secure against brute force using modern-day computing.

Table 2: Summary of Evaluated Algorithms

Cipher	Type of structure	No. of rounds	Key size (bits)	Operation mode
DES	Feistel Network	16	56	CBC
TDES	Feistel Network	48	168	CBC
AES	Substitution-Permutation Network	10	128	CBC
Blowfish	Feistel Network	16	32 to 448	CBC
Twofish	Feistel Network	16	128	CBC

Although TDES is quite adequate, there is a performance issue that makes the encryption/decryption process much slower than DES. So, NIST called for a proposal to develop a new algorithm that would overcome the drawbacks of TDES. Hence, an Advanced Encryption Standard (AES) was proposed, which uses the Rijndael algorithm. AES is part of many protocols and has been unbroken until now. AES is a symmetric block cipher that processes plaintext in 128-bit blocks. AES supports 10, 12, and 16 rounds of transformations for which the key size is 128, 192, and 256, respectively. We used AES with ten rounds and a 128-bit key for experimentation. The Blowfish algorithm is a symmetric block cipher proposed to overcome the weakness of DES. Blowfish processes the plaintext as 64-bit blocks. The key size is variable from 32 bits to 448 bits. Blowfish uses 16 rounds of operations to convert plaintext to ciphertext.

Twofish is a symmetric block cipher that has been in the race with the AES algorithm to become part of the standard. Twofish takes the input plaintext as 128-bit blocks. It passes the input message through 16 rounds of operations to convert it into ciphertext. Twofish supports variable key of sizes 128, 192, and 256 bits. A summary of the algorithms evaluated and the parameters chosen for evaluation are shown in Table. 2.

Measures and Metrics

The performance of the chosen algorithms was measured by considering the metrics encryption time, decryption time, throughput, battery consumption, CPU load and RAM utilization. The encryption time is the time to convert the input message to ciphertext. The decryption time is the time taken to convert

the ciphertext to the input message. The throughput is defined as the number of bytes processed per second. Battery consumption can be calculated using the formula given in Equation (1).

Battery Consumption (%) = (1/throughput) * (input_size) * 1000; (1)

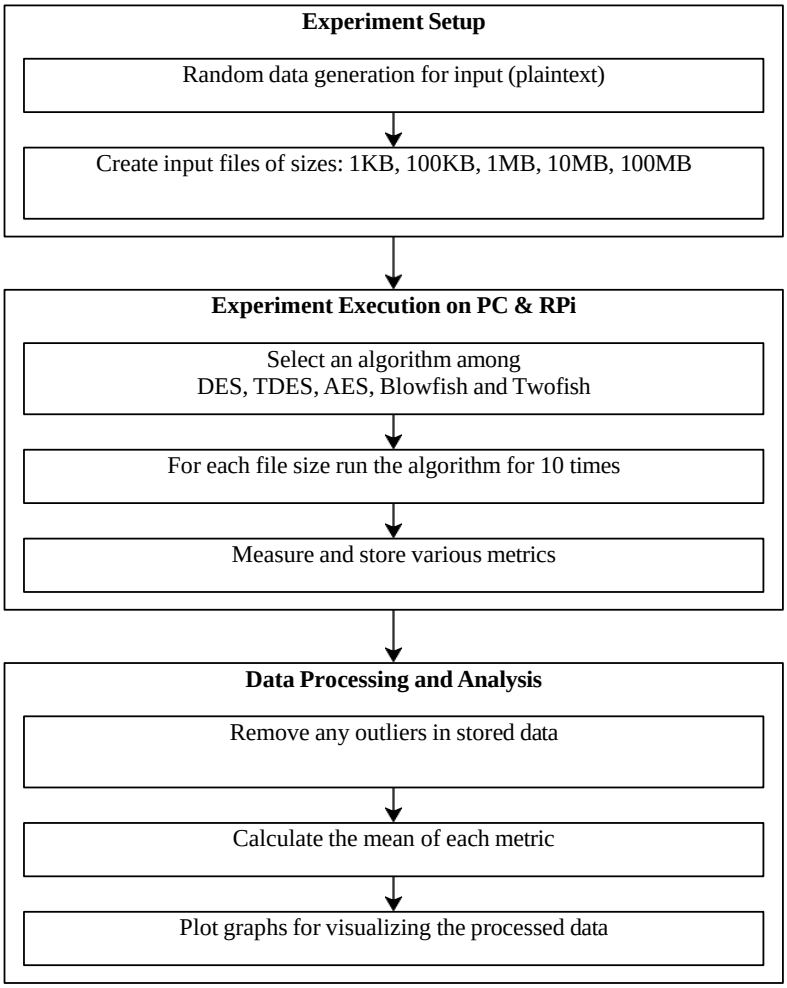

Figure1: Overview of Experimental Design and Methodology

The CPU load can be measured in terms of the load of running code on the CPU. Finally, the RAM utilization can be measured using the formula given in Equation (2).

Ram utilization (%) = (Memory used by the program / Total memory used by JVM) * 100 (2)

We considered the memory used by the Java Virtual Machine (JVM) in the above formula as the experimentation was done using Java programming language,

4 Methodology

The selected algorithms are evaluated on a high-end desktop machine and a low-end Raspberry Pi (RPi) with constrained resources. The RPi is part of an open testbed known as the FIT IoT lab that allows researchers worldwide to access IoT hardware and software free of cost (Adjih et al., 2015). The CPU power of the desktop PC is almost double the processing power of RPi. The algorithm implementation and evaluation was done using Java language version 19. Figure (1) presents an overview of the methodology followed for the experimentation of algorithms.

The specification of the desktop PC consists of an Intel Core i5 processor with a processing speed of 2.40 GHz and 16 GB RAM. The operating system on the PC is Windows 10 Pro. The specification of RPi, which is a part of the FIT IoT lab, consists of an ARM Cortex-A53 processor with a processing speed of 1.2 GHz and 1 GB RAM. The operating system used is a Linux distribution named Yocto. Java Cryptographic Extension (JCE), an official extension of Java, is used to evaluate cryptographic algorithms like DES, 3DES, AES, and Blowfish. Bouncy Castle, a third-party Java library, evaluates the Twofish algorithm.

The input data for evaluation is generated randomly and stored in various files of sizes 1KB, 100KB, 1MB, 10MB, and 100MB. Different file sizes helped to gauge the variation in the metrics for various algorithms. The experiments for evaluating the algorithms are carried out ten times per algorithm per file size. The purpose of repeating the experiments is to remove any outliers and ensure consistency among the readings. All the metrics are calculated and stored for later processing for each algorithm run. Each parameter's average or mean value is calculated for all the algorithms for later analysis.

5 Results and Discussion

This section presents and discusses the results of evaluating cryptographic algorithms like DES, TDES, AES, Blowfish, and Twofish on a desktop PC and on an RPi.

Results of Evaluation on PC

This section visualizes the results obtained by measuring encryption time, decryption time, throughput, battery life, CPU load, and RAM utilization for the algorithms on a desktop PC.

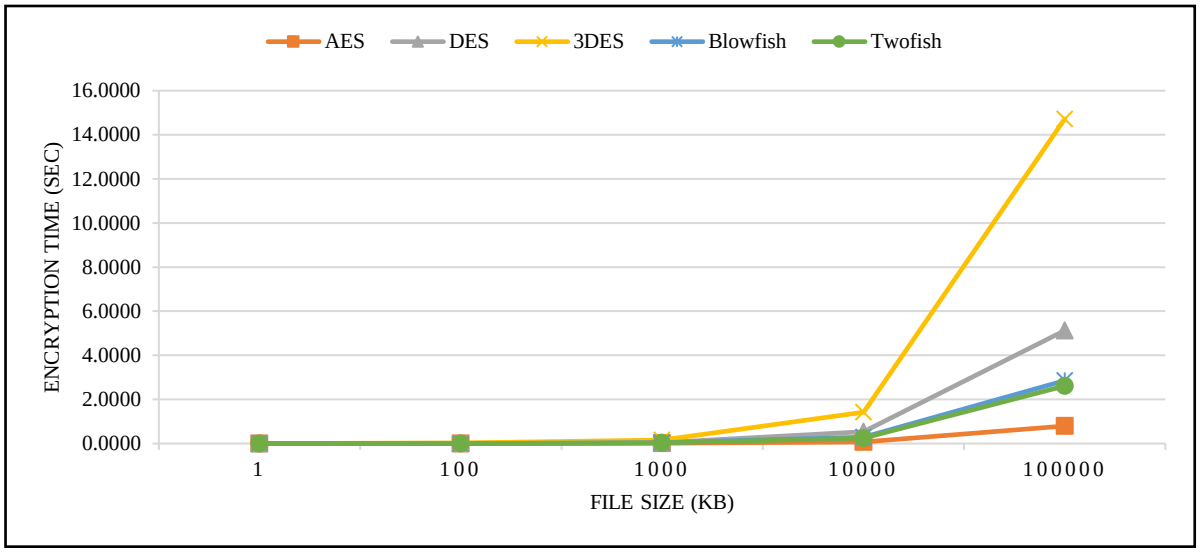

Figure 2a: Encryption Time on PC

Figure (2-a) shows that AES has the best encryption time across all file sizes, and TDES has the worst. For file sizes less than 10MB, the encryption time is almost identical for all algorithms. Figure (2-b) shows that AES has the best decryption time across all file sizes, and TDES has the worst decryption time. For a file size of 100MB, there is a slight variation in the decryption between Twofish and Blowfish, which is absent for encryption time. Figure (2-c) shows that AES has the best throughput across all file sizes, and TDES has the worst. It can also be seen that Twofish performs better than Blowfish in terms of throughput.

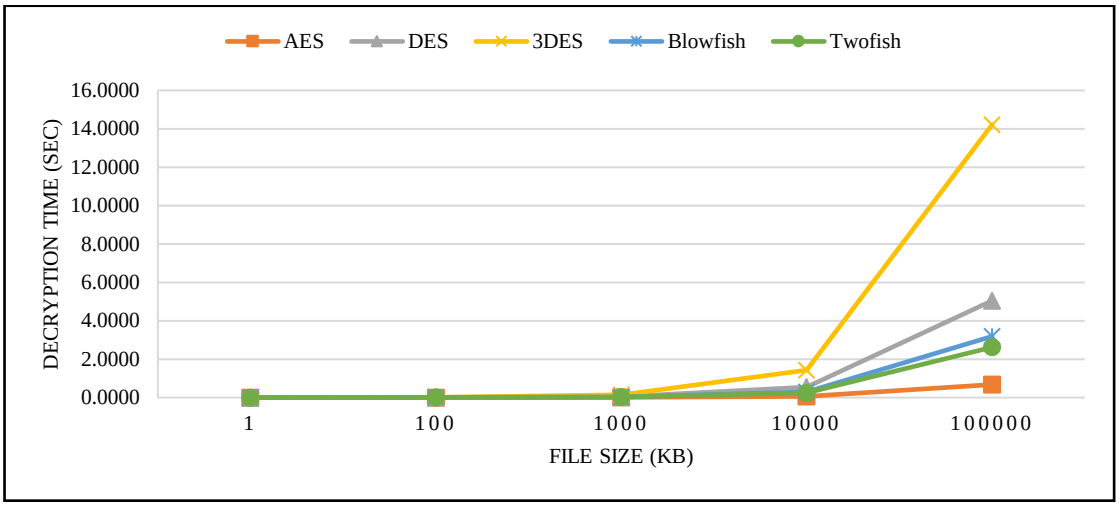

Figure 2b: Decryption Time on PC

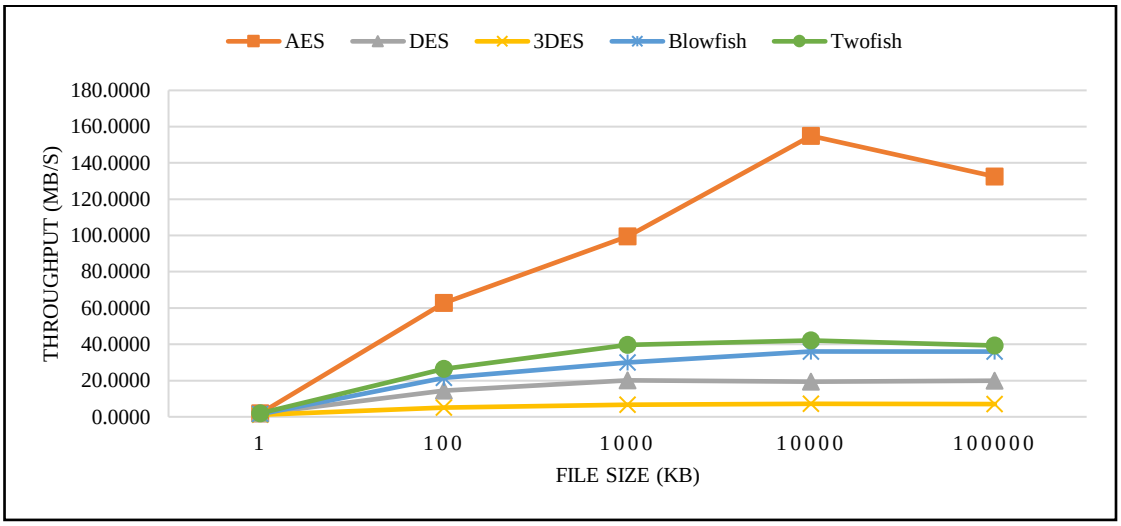

Figure 2c: Throughput on PC

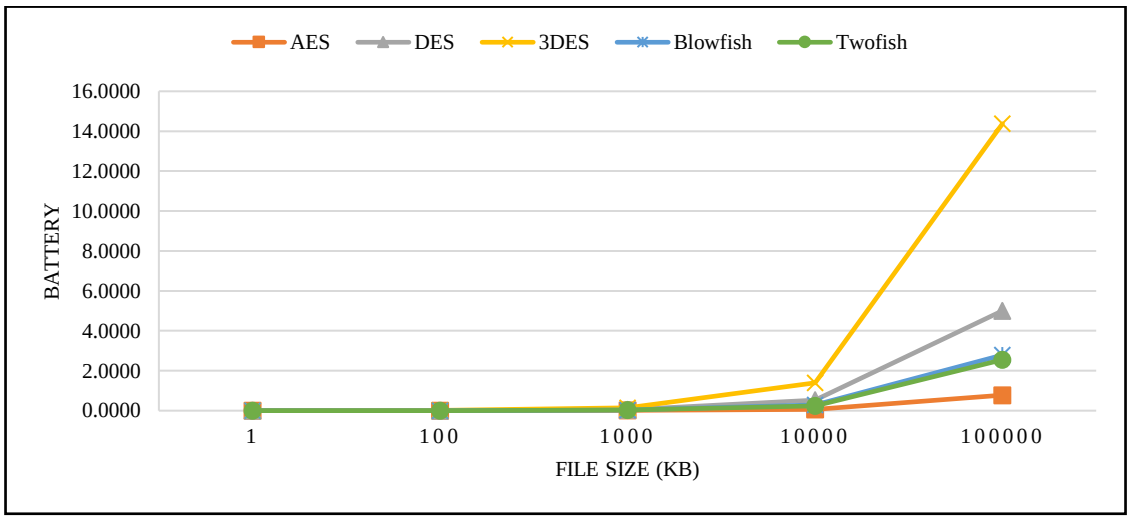

Figure 2d: Battery Consumption on PC

In Figure (2-d), it can be seen that AES has the least battery consumption while TDES has the highest battery consumption. In Figure (2-e), it can be seen that AES consumes the least CPU while TDES consumes the most CPU power for encryption and decryption. In Figure (2-f), it can be seen that RAM usage for AES is consistently high across all file sizes. For a file size of 10MB, there is a sharp increase in RAM usage for DES, TDES, and Blowfish algorithms. For file size of 100MB, there is a sharp decrease in RAM usage for DES and Twofish, while for TDES, it is constant.

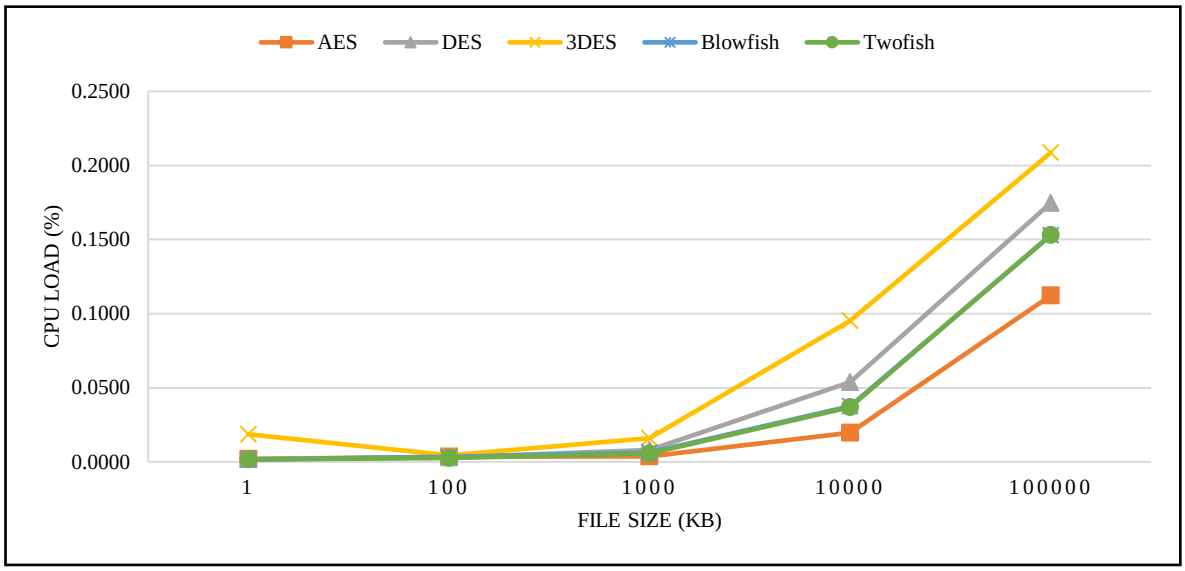

Figure 2e: CPU Load on PC

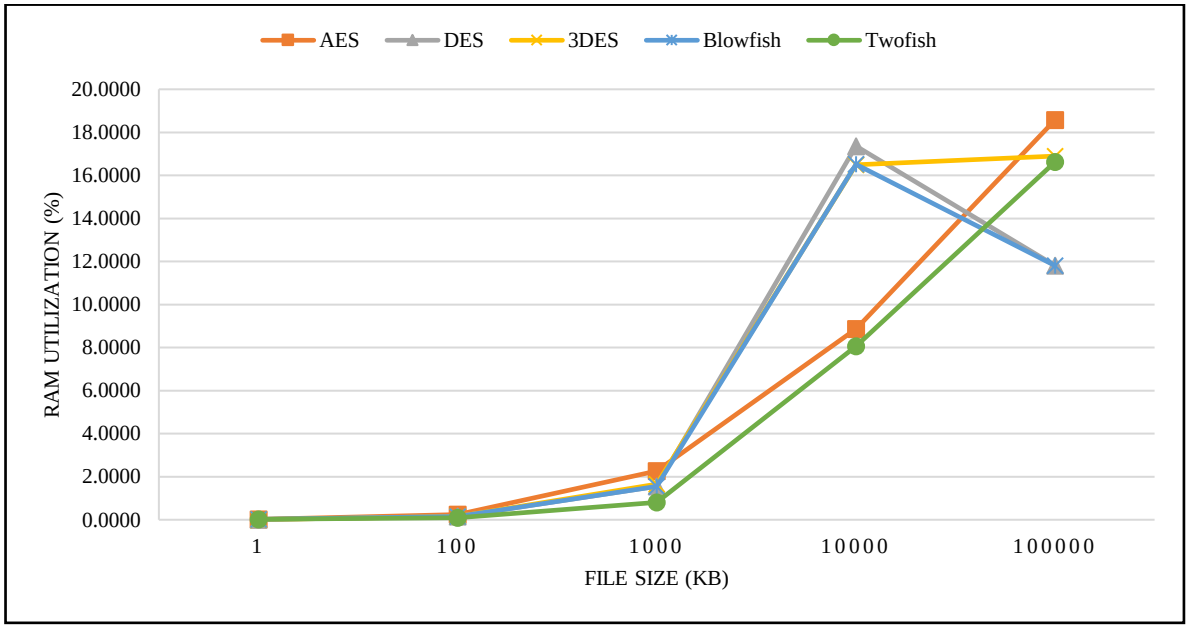

Figure 2f: RAM Utilization on PC

On average, AES performs better than Twofish across all the parameters except for RAM usage. The worst-performing algorithm is 3DES, followed by DES. Based on the experimentation carried out and the results, it is recommended to use AES or Twofish for data security on a PC, as RAM size is not a constraint.

Results of Evaluation on RPi

The results obtained by measuring encryption time, decryption time, throughput, battery, CPU load, and RAM utilization for the algorithms on an RPi are visualized in this section.

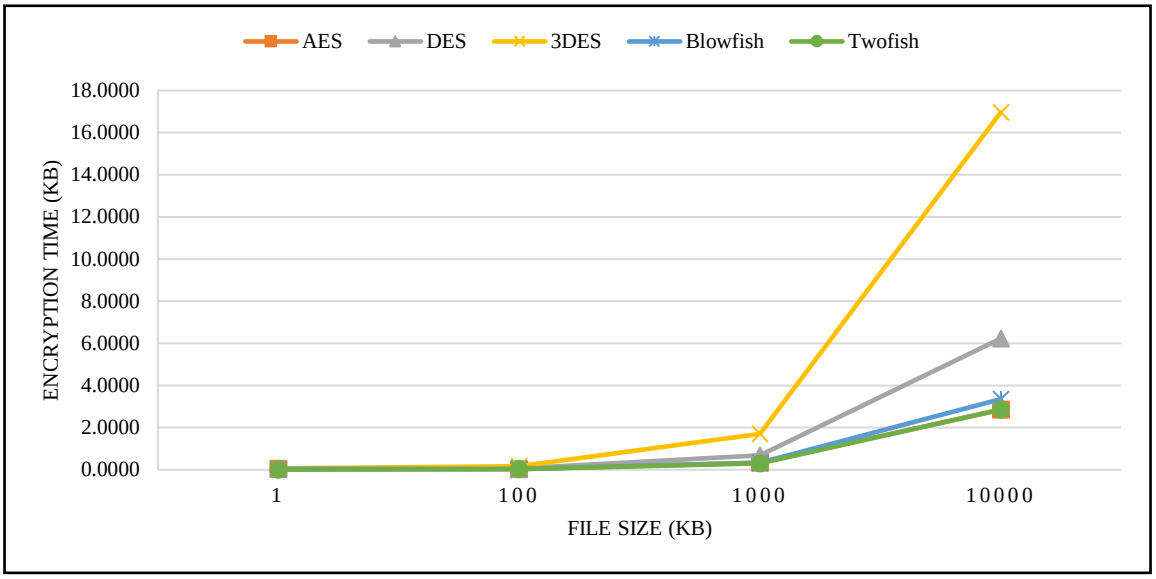

Figure 3a: Encryption Time on RPi

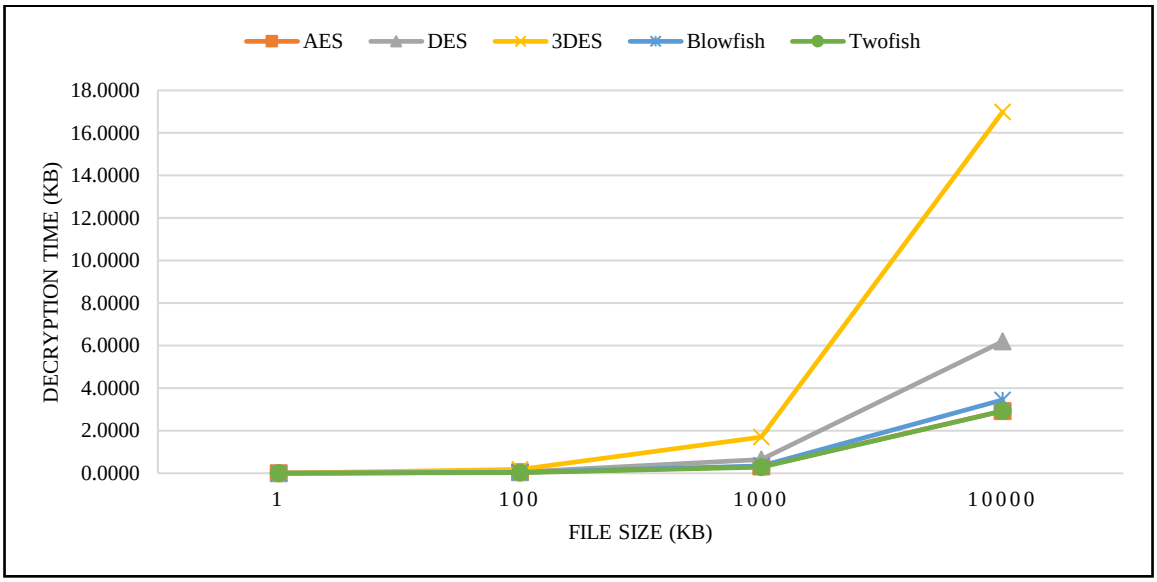

Figure 3b: Decryption Time on RPi

In Figure (3-a), it can be seen that AES, Blowfish, and Twofish have the best encryption time across all file sizes, and TDES has the worst. For file sizes less than 1MB, the encryption time is almost identical for all algorithms. In Figure (3-b), it can be seen that AES, Blowfish, and Twofish have the best decryption time across all file sizes, and TDES has the worst decryption time. The decryption time is almost the same for file sizes, which are less than 1MB for all algorithms. In Figure (3-c), it can be seen that Twofish has the best throughput for smaller file sizes, followed by AES, and TDES has the least throughput. It can also be seen that Twofish performs better than Blowfish in terms of throughput.

Figure (3-d) shows that AES, Blowfish, and Twofish have the least battery consumption, while TDES has the highest battery consumption. In Figure (3-e), it can be seen that AES consumes the least CPU while TDES consumes the most CPU power for encryption and decryption. In Figure (3-f), it can be seen that RAM usage for AES is consistently high across all file sizes. For a file size of 1MB, there is a sharp increase in RAM usage for DES, TDES, and Blowfish algorithms. For a file size of 10MB, there is a sharp decrease in RAM usage for DES, TDES, and Blowfish algorithms. One interesting thing is that Twofish consumes the least amount of RAM.

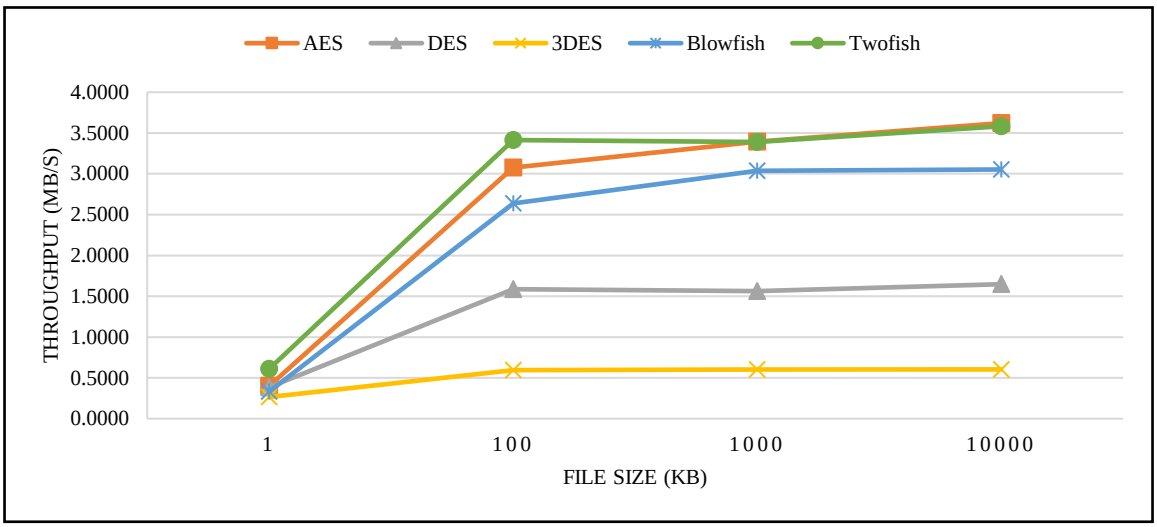

Figure 3c: Throughput on RPi

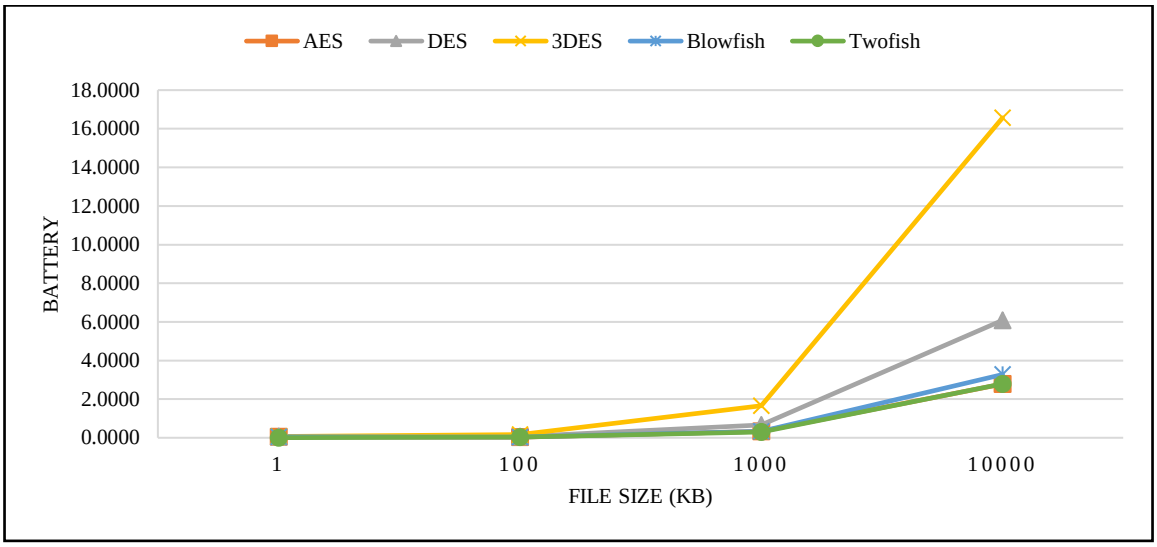

Figure 3d: Battery Consumption on RPi

On average, Twofish performance across all parameters is better than AES except for CPU usage. The worst-performing algorithm is 3DES, followed by DES. As per the experimentation and results, it is recommended to use Twofish for data security on IoT devices like RPi, as RAM size is a constraint in IoT devices. Finally, the evaluation of the performance of contemporary algorithms like DES, TDES, AES, Blowfish, and Twofish reveals that Twofish consistently delivers superior performance for end-to-end data encryption and decryption, proving its effectiveness across a wide range of devices, from high-end systems to low-end IoT devices.

Now, we discuss the analysis of each cipher considered in this evaluation concerning security, performance, and resource efficiency. The security of DES is weak as the key size is only 56 bits. A brute force attack can yield the key within a small amount of time by using modern-day computers. The performance of DES is low as it takes more time to encode and decrypt, thereby affecting the throughput. The resource efficiency of DES is low and is a bad choice for IoT devices compared to other ciphers. The security of 3DES is not the best. Because of poor cipher design, attacks like the Meet-In-The-Middle and birthday attacks can yield the key. The performance of 3DES is low and is even poor when compared with DES. The resource efficiency of 3DES is not the best. RAM utilization of 3DES is better than that of DES.

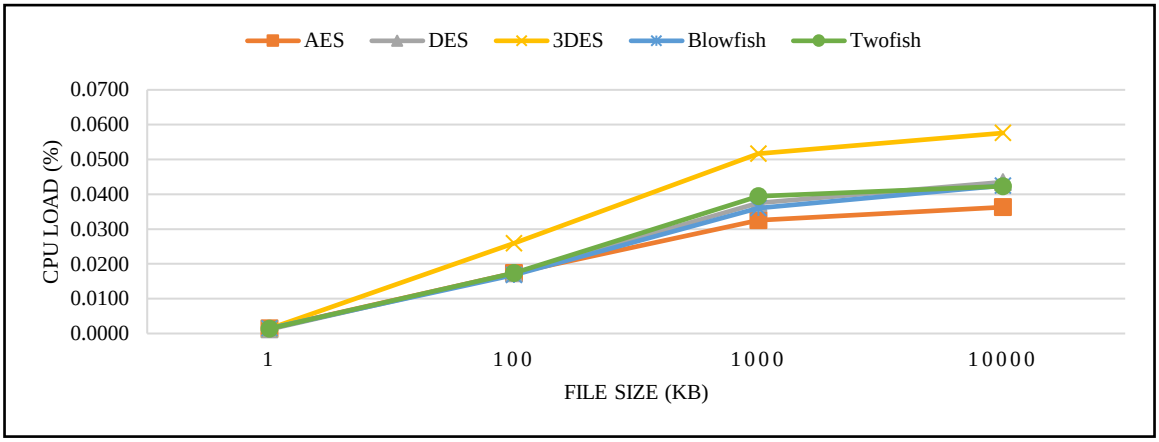

Figure 3e: CPU Load on RPi

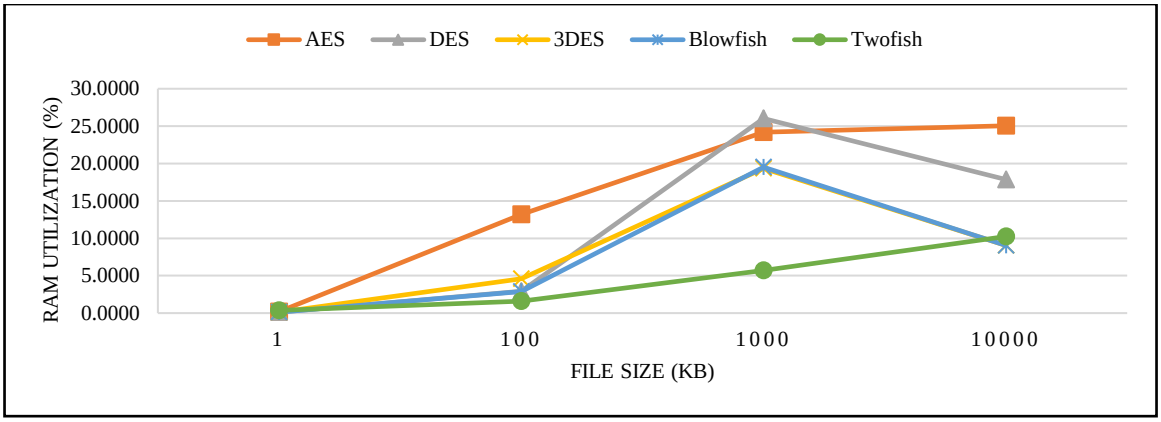

Figure 3f: RAM Utilization on RPi

The security of AES is strong as the key size is 128 bits and due to its strong design. The performance of AES is high, and the throughput is better. The resource efficiency of AES is relatively low as the RAM usage is quite high and may not be suitable for IoT devices. The security of Blowfish is strong due to its variable key size of up to 448 bits. The performance of Blowfish is high and is similar to AES. The resource usage of Blowfish is high, and its RAM usage is less than AES, which makes it a suitable candidate for IoT devices. The security of Twofish is strong due to its key size of 128 bits and its cipher design. The performance of Twofish is high and is even better than Blowfish. The resource usage of Twofish is high, and even better than Blowfish in terms of RAM utilization. So, based on the evaluation results, Twofish is the best cipher in terms of security, performance, and resource efficiency among all the ciphers considered in this evaluation.

6 Conclusions and Future Scope

This work evaluates the performance of popular algorithms like DES, TDES, AES, Blowfish, and Twofish, generally used for encryption and decryption on powerful systems. The performance of these algorithms is evaluated on a high-end desktop PC and low-end RPi, which is part of the FIT IoT lab open testbed. We can conclude from the experimental results on PC that AES offers better performance than Twofish. We can conclude from the experimental results on RPi that Twofish performs better than AES. Finally, for end-to-end encryption and decryption in IoT networks, we suggest that Twofish is a better choice as a whole, as seen from the experimental results. Due to various reasons, we could not procure similar high-end IoT devices like ESP32-WROOM, Banana Pi MP3, Libre Computer Board, and BeagleBone Black, which are widely used as controllers in many IoT projects and systems besides RPi. Hence, the future scope for this work includes evaluating the performance of these algorithms on these devices. We expect similar results on these devices, too.

References

[1] Abd Elminaam, D. S., Abdual-Kader, H. M., & Hadhoud, M. M. (2010). Evaluating the performance of symmetric encryption algorithms. *International Journal of Network Security, 10*(3), 213-219.

[1] Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., ... & Watteyne, T. (2015, December). FIT IoT-LAB: A large scale open experimental IoT testbed. *In 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (pp. 459-464). IEEE. https://doi.org/10.1109/WF-IoT.2015.7389098

[2] Al Tamimi, A. K. (2006). Performance analysis of data encryption algorithms. *available at weblink: http://www. cse. wustl. edu/~ jain/cse567-06/ftp/encryption_perf/index. html*

[3] Aswathy, S. (2024). Bibliometric Analysis of Sustainability in Business Management Policies Using Artificial Intelligence. *Global Perspectives in Management, 2*(1), 44-54.

[4] Centeno, J. K. M., Chhabra, P. S., Fianza, C. L., Montes-Austria, I., & Ocampo, R. (2018, October). Performance Analysis of Encryption Algorithms on Smartwatches. In *TENCON 2018-2018 IEEE Region 10 Conference* (pp. 0162-0166). IEEE. https://doi.org/10.1109/TENCON.2018.8650067

[5] El-Haii, M., Chamoun, M., Fadlallah, A., & Serhrouchni, A. (2018, October). Analysis of cryptographic algorithms on iot hardware platforms. In *2018 2nd Cyber Security in Networking Conference (CSNet)* (pp. 1-5). IEEE. https://doi.org/10.1109/CSNET.2018.8602942

[6] El-Hajj, M., Mousawi, H., & Fadlallah, A. (2023). Analysis of lightweight cryptographic algorithms on iot hardware platform. *Future Internet*, *15*(2), 54. https://doi.org/10.3390/fi15020054

[7] Elminaam, D. S. A., Kader, H. M. A., & Hadhoud, M. M. (2009). Performance evaluation of symmetric encryption algorithms on power consumption for wireless devices. *International Journal of Computer Theory and Engineering, 1*(4), 343–351. https://doi.org/10.7763/IJCTE.2009.V1.54

[8] Farhan, L., Shukur, S. T., Alissa, A. E., Alrweg, M., Raza, U., & Kharel, R. (2017, December). A survey on the challenges and opportunities of the Internet of Things (IoT). In *2017 Eleventh International Conference on Sensing Technology (ICST)* (pp. 1-5). IEEE. https://doi.org/10.1109/ICSensT.2017.8304465

[9] Farooq, M., Waseem, M., Mazhar, S., Khairi, A., Kamal, T., Khan, R., Khan, S. U., Zaheer, R., Khan, S., Shen, G., Liu, B., Yang, D., Liu, F., Guillemin, P., Friess, P., Woelffl, S., Hauben, R., An, J., Gui, X., & He, X. (2015). A review on internet of Things (IoT). *International Journal of Computer Applications in Technology, 113*(1), 1–7.

[10] Fotovvat, A., Rahman, G. M., Vedaei, S. S., & Wahid, K. A. (2020). Comparative performance analysis of lightweight cryptography algorithms for IoT sensor nodes. *IEEE Internet of Things Journal*, *8*(10), 8279-8290. https://doi.org/10.1109/JIOT.2020.3044526

[11] Haque, M. E., Zobaed, S. M., Islam, M. U., & Areef, F. M. (2018, December). Performance analysis of cryptographic algorithms for selecting better utilization on resource constraint devices. In *2018 21st International Conference of Computer and Information Technology (ICCIT)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICCITECHN.2018.8631957

[12] Hassan, W. H. (2019). Current research on Internet of Things (IoT) security: A survey. *Computer networks*, *148*, 283-294. https://doi.org/10.1016/j.comnet.2018.11.025

[13] Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A survey on IoT security: application areas, security threats, and solution architectures. *IEEe Access*, *7*, 82721-82743. https://doi.org/10.1109/ACCESS.2019.2924045

[14] Hossain, M. A., Hossain, M. B., Uddin, M. S., & Imtiaz, S. M. (2016). Performance analysis of different cryptography algorithms. *International Journal of Advanced Research in Computer Science and Software Engineering*, *6*(3), 659-665.

[15] Kansal, S., & Mittal, M. (2014, December). Performance evaluation of various symmetric encryption algorithms. In *2014 international conference on parallel, distributed and grid computing* (pp. 105-109). IEEE. https://doi.org/10.1109/PDGC.2014.7030724

[16] Kavitha, M. (2024). Environmental Monitoring Using IoT-Based Wireless Sensor Networks: A Case Study. *Journal of Wireless Sensor Networks and IoT*, *1*(1), 32-36.

[17] Ofoghi, R. (2015). Technical and structural analysis of the wireless networks, safety and security analysis of wireless and cable networks. *International Academic Journal of Science and Engineering, 2*(1), 39–44.

[18] Ojo, M. O., Giordano, S., Procissi, G., & Seitanidis, I. N. (2018). A review of low-end, middle-end, and high-end IoT devices. *IEEE Access*, *6*, 70528-70554. https://doi.org/10.1109/ACCESS.2018.2879615

[19] Padmavathi, B., & Kumari, S. R. (2013). A survey on performance analysis of DES, AES and RSA algorithm along with LSB substitution. *IJSR, India*, *2*(4), 170-174.

[20] Papadopoulos, G., & Christodoulou, M. (2024). Design and Development of Data Driven Intelligent Predictive Maintenance for Predictive Maintenance. *Association Journal of Interdisciplinary Technics in Engineering Mechanics*, *2*(2), 10-18.

[21] Purnama, Y., Asdlori, A., Ciptaningsih, E. M. S. S., Kraugusteeliana, K., Triayudi, A., & Rahim, R. (2024). Machine Learning for Cybersecurity: A Bibliometric Analysis from 2019 to 2023. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *15*(4), 243-258. https://doi.org/10.58346/JOWUA.2024.I4.016

[22] Renner, S., Pozzobon, E., & Mottok, J. (2019). Benchmarking software implementations of 1st round candidates of the NIST LWC project on microcontrollers. *National Institute of Standards and Technology (NIST)*.

[23] Rizvi, S. A. M., Hussain, S. Z., & Wadhwa, N. (2011, June). Performance analysis of AES and TwoFish encryption schemes. In *2011 International Conference on Communication Systems and Network Technologies* (pp. 76-79). IEEE. https://doi.org/10.1109/CSNT.2011.160

[24] Saraiva, D. A., Leithardt, V. R. Q., de Paula, D., Sales Mendes, A., González, G. V., & Crocker, P. (2019). Prisec: Comparison of symmetric key algorithms for iot devices. *Sensors*, *19*(19), 4312. https://doi.org/10.3390/s19194312

[25] Shah, S. H., & Yaqoob, I. (2016). A survey: Internet of Things (IOT) technologies, applications and challenges. *2016 IEEE Smart Energy Grid Engineering (SEGE)*, 381-385. https://doi.org/10.1109/SEGE.2016.7589556

[26] Shah, Y., & Sengupta, S. (2020, October). A survey on Classification of Cyber-attacks on IoT and IIoT devices. In *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)* (pp. 0406-0413). IEEE. https://doi.org/10.1109/UEMCON51285.2020.9298138

[27] Sharif, S. O., & Mansoor, S. P. (2010, August). Performance analysis of stream and block cipher algorithms. In *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)* (Vol. 1, pp. V1-522). IEEE. https://doi.org/10.1109/ICACTE.2010.5578961

[28] Sreevidya, B., & Supriya, M. (2024). Trust based Routing – A Novel Approach for Data Security in WSN based Data Critical Applications. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 15*(1), 27-41. https://doi.org/10.58346/JOWUA.2024.I1.003

[29] Suryateja, P. S., & Rao, K. V. (2024). A Survey on Lightweight Cryptographic Algorithms in IoT. *Cybernetics and Information Technologies, 24*(1), 21-34. https://doi.org/10.2478/cait-2024-0002

[30] Thakor, V. A., Razzaque, M. A., & Khandaker, M. R. (2021). Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities. *IEEE Access, 9*, 28177-28193. https://doi.org/10.1109/ACCESS.2021.3052867

[31] Umaparvathi, M., & Varughese, D. K. (2010, December). Evaluation of symmetric encryption algorithms for MANETs. In *2010 IEEE International Conference on Computational Intelligence and Computing Research* (pp. 1-3). IEEE. https://doi.org/10.1109/ICCIC.2010.5705754

[32] Verma, H. K., & Singh, R. K. (2012). Performance analysis of RC6, Two fish and Rijndael block cipher algorithms. *International Journal of Computer Applications, 42*(16), 1-7.

[33] Vyakaranal, S., & Kengond, S. (2018, April). Performance analysis of symmetric key cryptographic algorithms. In *2018 international conference on communication and signal processing (ICCSP)* (pp. 0411-0415). IEEE. https://doi.org/10.1109/ICCSP.2018.8524373

Authors Biography

P.S. Suryateja, M.Tech., is a Research Scholar (full-time Ph.D.) in the Dept. of Computer Science & Systems Engineering at Andhra University, Visakhapatnam. He previously worked as an associate professor in the CSE department at Vishnu Institute of Technology, India. He has 13+ years of teaching experience and is an individual researcher whose research interests are Cloud Computing, Internet of Things, Computer Security, Network Security and Blockchain. He is a member of professional societies such as IEEE, ACM, CSI, and ISCA. He published several research papers, which have been indexed by SCIE, WoS, Scopus, Springer, and others.

Dr. Kasukurthi Venkata Rao, a professor in the Department of Computer Science and Systems Engineering, is currently the head of the largest department at the University. He holds vast teaching experience spanning 28+ years. Dr. Rao's research potential is noteworthy, with 25 scholars under his guidance. He published several research papers, which have been indexed by SCIE, WoS, Scopus, Springer, and others. His research interests mainly focus on medical image processing and analysis and network issues using state-of-the-art machine learning technologies. To add impetus to his academic credentials, he has undergone training for quality improvement in education at NIT-W, XIME-B, JNTUA, Infosys, IIM-Indore, and others.