TIERS Information Technology Journal
Vol.5, No.2, December 2024, pp. 184~192
P-ISSN: 2723-4533 / E-ISSN: 2723-4541
DOI: https://doi.org/10.38043/tiers.v5i2.6039

Experimental Comparison of Encryption Algorithms On Smart Devices

Qurban Ali Frugh[1], Mohammad Fahim Naseri[2], Musawer Hakimi[3]
qurbanalifru@gmail.com[1], fahim.naseri4@gmail.com[2] , musawer@adc.edu.in[3]
[1,2]Department of Information Technology, Computer Science Faculty, Kabul Education University, Afghanistan
[3]Department of Computer Science, Samangan University, Afghanistan

ABSTRACT

Now, the technological environments rely on a large number of users and devices interconnected in such a way as to be able to share information and exchange resources. For such reasons, security becomes of prime importance inside these networks. Especially, encryption algorithms widely used in smart metering systems form the very backbone of ensuring security. It is unique according to every single parameter: the level of security achieved, speed, operational complexity, length, and type of key used. A comparison of performance and throughput for the most used encryption algorithms, such as AES-128, AES-192, RC4, Blowfish, and ECDSA, is presented here. Such devices, for instance, smart meters, usually represent very resource-constrained computational capability, memory, and data transfer. For these reasons, the experiments investigate performance impacts of using different encryption algorithms on a smart meter environment in a scaled setup. Experiments were run on a laptop device and then downscaled according to the limitation characteristics of the smart meter target device. As for the scaling, it has been performed concerning key factors: CPU of the smart meter, RAM, and cache memory. Execution times of the encryption and decryption processes were measured, as well as the throughput of messages for various file sizes. Quantitative key results obtained included: RC4 with a throughput of 25.37 Kbytes/sec, whereas AES-128 has 4.87 Kbytes/sec, while in ECDSA-256 the performance is much lower: its verification throughput amounts to 0.0023 Kbytes/sec. The results showed significant variations in performance, speed, and throughput between these algorithms. Even considering small smart devices, the encryption and decryption processes yielded efficient throughputs. These findings underscore the importance of choosing the right encryption algorithm for smart meters, balancing both security requirements and resource limitations to ensure optimal performance.

Keywords: encryption algorithms; smart devices; performance; throughput; comparison.

Article Info

Accepted	:	01-11-2024
Revised	:	02-09-2024
Published Online	:	25-12-2024

Correspondence Author:

Qurban Ali Frugh
Information Technology Department,
Kabul Education University, Kabul, Afghanistan
Email: qurbanalifru@gmail.com

1. INTRODUCTION

With the rapid advance of technology, especially the rapid growth in volume of digital information, more serious challenges are raised against effective secure data exchange over the computer networks [1]. Due to the increase in data sharing within various applications, especially those that concern communication networks and involve more users interacting, information security has been at high risk. Encryption algorithms are vital in protecting data from unauthorized access, but they differ in several aspects regarding speed, security strength, and computational efficiency [2]. While some encryption algorithms offer strong security, they are computationally expensive, hence unsuitable for resource-constrained devices. Others are efficient but may not offer strong security features, thus making systems prone to attacks [3]

In recent years, mobile phones, smart home appliances, medical sensors, and IoT-enabled industrial systems are being widely adopted. Most of these devices have weak processing power, memory, and energy resources, which again restricts them to implement complex cryptographic algorithms [4]. So, choosing an appropriate encryption algorithm for IoT applications has turned out to be a very critical research area. Although several works have performed the comparison among different encryption methods, most of them have not considered real-world device constraints, algorithm performance, and security trade-offs [5]

This research paper tries to fill this gap by systematically investigating and comparing several encryption algorithms, including AES, 3DES, RC4, Blowfish, ECDSA, and Camellia128 with different key lengths. The paper uses the OpenSSL security performance and efficiency testing tool to measure the encryption speed, computational overhead, and overall effectiveness in securing smart devices. These empirical performance evaluations performed in this work will definitely provide insight into the selection of the most suitable encryption algorithms for specific IoT applications. The novelty of the research presented herein lies in that it is oriented to real-world implementation scenarios [6-10].

It contributes by investigating the processing capabilities of the IoT devices in light of encryption overhead and energy consumption. All these aspects are of importance when it comes to ensuring security in smart environments while balancing performance with protection of data. Conclusively, this study will provide added value in the field of information security as it will outline practical recommendations regarding the choice of an encryption algorithm that can enhance security without decreasing device efficiency. Since IoT technologies are increasingly becoming applied in smart homes, health care, and industrial automation, securing these with efficient cryptographic solutions is gaining even more importance [12-16].

The findings from this research would guide cybersecurity experts, system architects, and developers of IoT items to implement good yet efficient mechanisms of encryption. This will hopefully make smart gadgets resilient against every kind of cyber threat.

State Of The Art

For data communication, cryptography encompasses various methods designed to protect communication between the sender and the receiver. The word "cryptography" is derived from two Greek words: "kryptos," meaning "hidden," and "graphein," meaning "to write." Essentially, cryptography is the science of writing in a hidden manner, which includes the processes of text and message encryption and decryption [2][17].

Encryption is divided into two important and separate operations: the first is called encryption, and the second is called decryption [3][14]. The encryption process converts plaintext (the original message) into ciphertext (the hidden message), which cannot be read by unauthorized users, while the decryption process retrieves the original message from the hidden message [2] [8-10]. As mentioned in the title introducing the topic, cryptographic algorithms are classified into two main sections: symmetric algorithm encryption and asymmetric algorithm encryption. This classification is illustrated in the figure below:

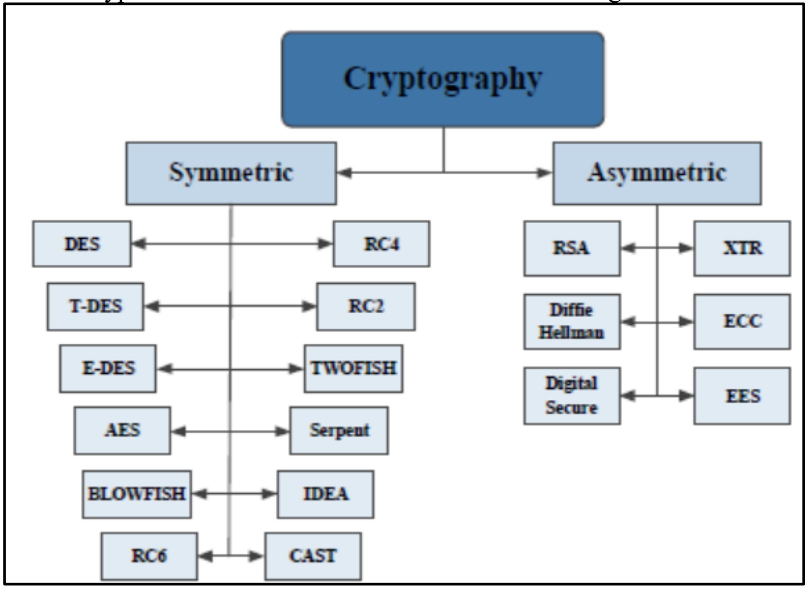

Figure 1: Classifications of Cryptographic Algorithms [2]

In many articles, the RSA algorithm from the category of asymmetric algorithms and the AES algorithm from the category of symmetric algorithms have been compared as examples for information encryption [2].

Table 1: Comparison of AES and RSA [2]

Security/ performance	Excellent	Excellent
Flexibility	Yes	No
Structure	Substitution permutation	Public key algorithm
Key Size (bit)	128, 92, 256	1024, 4096
Block size(bits)	128	128
Year	2001	1978
Items	AES	RSA

Therefore, in this article, the review of these two algorithms has been given more attention. the RSA and AES encryption algorithms are compared based on the year of introduction, creator, data block size, key size, structure, flexibility, and other capabilities. Evaluation of RSA and AES algorithms on different word counts was performed on an Intel-R core-tm with specifications of the core i-7 model 4510U, a 2.6GHz processor, and a 64-bit-based process with 12GB RAM. Security analysis in this experiment has been tested with respect to encryption and decryption time. The time required from the stage of converting plaintext to ciphertext is defined as the encryption time of the algorithm. The time taken to decrypt the encrypted text is defined as the message decryption time [2][15].

The table below shows the difference in encryption time for messages of various word lengths between the RSA and AES algorithms. The execution time of encryption in the RSA and AES algorithms is shown in Table 2 as follows.

Table 2: Compare of execution time RSA & AES [2] [8].

No. of Words	AES	RSA
100	0.41271	1.8438
200	0.81641	3.5367
400	1.5961	7.0457
800	3.2224	14.1674
1000	4.037	18.5096
2000	8.0976	34.7036
5000	20.282	86.7242

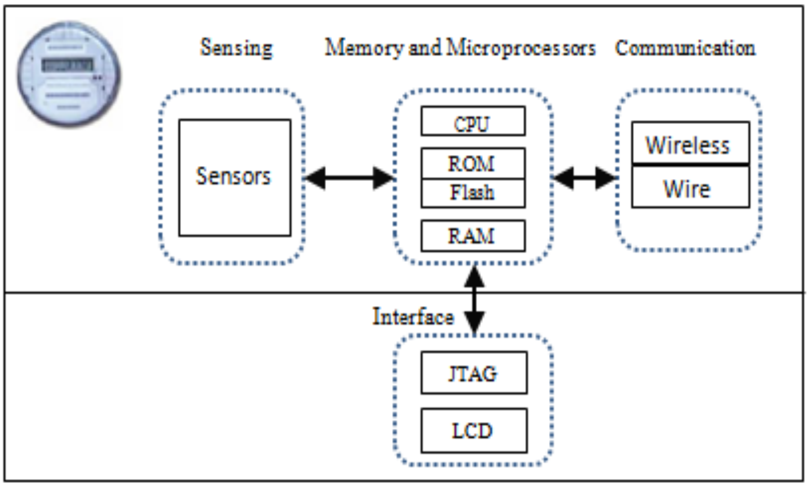

Figure 2. Smart meter architecture [10]

Research [6] has surveyed several proposed mechanisms regarding symmetric algorithms and ultimately provided the basis for a comparative study.

.

Table 3: Comparison symmetric and asymmetric [3]

Parameter	Symmetric Encryption				Asymmetric Encryption	
	DES	3DES	AES	Blowfish	RSA	Diffie-Hellman
Key Used	Same key	Same key	Same key	Same key	Different keys	Key Exchange
Throughput	Lower than AES	Lower than AES	Lower than Blowfish	Very High	Low	Lower than RSA
Encryption Ratio	High	Moderate	High	High	High	High
Tunability	No	No	No	Yes	Yes	Yes
Power Consumption	Higher than AES	Higher than AES	Higher than Blowfish	Very Low	High	Lower than RSA
Key Length	56 Bits	112 to 168 Bits	128, 192, 256 Bits	32 Bit to 448 Bit	>1024 Bit	Key Exchange Management
Speed	Fast	Fast	Fast	Fast	Fast	Slow
Security Against Attack	Brute Force Attack	Brute Force-Plaintext, Known Plaintext	Chosen- Plain, Known Plaintext	Dictionary Attack	Timings Attacks	Eavesdropping

This article discusses the fundamental features, advantages, disadvantages, and various applications of symmetric encryption algorithms. The summary of all the aforementioned research is presented in the table 3.

A. Smart meter architecture

Smart meters are using AES algorithms for security and data security communication. smart meters are included three parts: the first. Microcontrollers with 8, 16, or 32-bit microprocessors, microcontrollers such as M series CPUs, CS7401xx logical series, which have RAM and flash memory. The second. Communication unit and third one is sensor unit, which is a smart meter and is designed for measuring energy consumption. This architecture is shown in Figure 2 [5] [9][19].

B. The limitations of smart meters

Smart meters have certain limitations that pose challenges in implementing any encryption algorithm, and in some cases, it is even impossible to implement any encryption algorithm. In general, smart meters are built on microprocessors that have memory and computational capacity limitations. For example, the CS7401xx microcontroller unit, which includes an ARM7TDMI™ 16/32-bit RISC processor unit, has 32kB - 128kB of flash memory and 8KB of chip RAM. The CS7401xx microcontroller unit includes a JTAG interface for debugging after the production [13][18]. This limitations in memory and weak computational ability in smart meters are challenges in implementing any encryption algorithms [9][20].

This For data communication, cryptography encompasses various methods designed to protect communication between the sender and the receiver. The word "cryptography" is derived from two Greek words: "kryptos," meaning "hidden," and "graphein," meaning "to write." Essentially, cryptography is the science of writing in a hidden manner, which includes the processes of text and message encryption and decryption [2][21].

2. RESEARCH METHOD

Given the goal of the research, the performance and effectiveness of cryptographic algorithms have to be experimented with and evaluated on specific devices. In order to do this, different methods of encryption were tested on high-performance and low-performance devices, which allowed a comparative analysis concerning speed, efficiency, and computational overhead. Careful attention has been paid in the scaling process to assess how encryption algorithms will perform under different hardware constraints. First, the testing of encryption algorithms was conducted in high-performance computing environments with advanced processors and huge memory. Later, the same tests were executed on low-performance IoT devices, such as microcontrollers and embedded systems, in order to analyze the degradation in performance and efficiency. In this way, one can get a practical view of the trade-offs involved in implementing cryptographic techniques in resource-constrained environments.

Each encryption algorithm was repeated 50 times under the same conditions in order to make the results reliable. Such repetition is quite essential for capturing statistical variations, smoothing out random errors, and hence increasing the accuracy of performance measurements. This is important for the research because with multiple runs, the reported times for encryption and decryption would be immune from transient system fluctuations, background processes, or network latencies. Moreover, these were averaged out across the different iterations so that more stable and representative results about each algorithm performance were obtained.

Many ways of validation are tried for ascertaining accuracy and repeatability for test results drawn through experiments: detection of outliers with a view to finding and deleting those that were potentially able to distort the readings; the result thus drawn, through both MS Excel and SPSS, had also been counter-checked using OpenSSL's built-in cryptographic performance tester, independent, third-party performance benchmarking applications. Third, statistical significance tests were conducted, including standard deviation and confidence interval analysis, to ensure the reliability of the data. These validation measures enhance the credibility of the findings and reinforce the contributions of this study in cryptographic performance analysis.

Systematic scale-up, repeated trials, and validation of experimental data present a comprehensive and empirically valid assessment of various encryption algorithms in high- and low-performance scenarios. The results offer practical recommendations on the choice of the best cryptographic solutions, considering the computation capability of the IoT device to improve the security of real-world data.

3. RESULTS AND DISCUSSION

Comparative and experimental section

To adapt this experience, the Benchmark performance and efficiency test of the cryptographic algorithms called OpenSSL has been used in the CLI environment. Therefore, the encryption and decryption process in various algorithms has been experienced using the good OpenSSL library. This experience has been such that specific information of a specific size has been encrypted and decrypted. To ensure the accuracy of encryption and decryption time, the number of operations has been repeated at least 50 times, and the average of these repetitions has been taken. In this experiment, the OpenSSL benchmark with version 3.4.0 in a Windows environment has been used to obtain the encryption and decryption times of various symmetric algorithms using text files. The speed and execution time of encryption and decryption were performed on a computer device with a 2GHz processor and 4GB of memory, and scaled to the speed of small computing devices such as smart meters with a 160MHz processor.

First experimental question: In a small computing device such as smart electricity meters, how do the sign time and verify time in the ECDSA algorithm vary with different key lengths, and how can we understand the difference? Therefore, the encryption and decryption time of these messages has been compared to evaluate the performance of the ECDSA algorithm with different key lengths. The size and volume of these messages are considered in the range of [4]. This experiment was conducted on a single-core 2 GHz processor.

Table 6: ECDSA Performance with different Key size- 2GHz Single-Core Processor.

		ECDSA Performance Table on a 2GHz Processor		
Key(bits) & type	sign(sec)	verify(sec)	sign/s	verify/s
160 (secp160)	0.0003	0.0009	3883.8	1054.7
192 (nistp192)	0.0003	0.0013	3175.6	798.9
224 (nistp224)	0.0004	0.0017	2493.4	592.8
256 (nistp256)	0.0005	0.0022	2012.6	463.7
384 (nistp384)	0.001	0.005	980.5	198.8
521 (nistp521)	0.002	0.0107	502.2	93.9

The following results, according to the above table, were obtained after scaling on a 160 GHz processor (smart electricity meters).

Table 7: ECDSA Performance Evaluation Table with Different Key Lengths Based on a 160 MHz Processor.

		ECDSA Performance Table on a 160 MHz CPU		
key(bit)& types	sign(sec)	verify(sec)	sign/s	verify/s
160 (secp160)	0.0038	0.011	303.422	82.398
192 (nistp192)	0.0038	0.016	248.093	62.414
224 (nistp224)	0.0051	0.021	194.797	46.312
256 (nistp256)	0.0064	0.028	157.234	36.227
384 (nistp384)	0.0128	0.064	76.602	15.531
521(nistp521)	0.0256	0.137	39.234	7.336

.

Comparison of the results with the method [8] regarding the delay for AES-128 and ECDSA-256 has been obtained with the average decryption in Table 5-3. It is worth noting that the throughput of each algorithm has been calculated [4].

Throughput=Plaintext (MB)/Encrypt/Decrypt (Sec.) …………………… (1)

Second experimental question: What is the throughput of different algorithms in one second? This issue was obtained by conducting various experiments with repeated processes on a 2GHz processor and has been presented in the table below after scaling.

Table 8: Throughput of different Algorithms

Num	Algorithms	Throughput (Kbytes/Sec)
1	RC4	25.37378311
2	Blowfish	5.593592834
3	AES 128	4.869560242
4	AES 192	4.182707214
5	DES	3.458629608
6	3DES	1.280587006
7	Camellia 128	6.221892548
8	ECDSA 192 Verify	0.003899815
9	ECDSA 256 Verify	0.002263415

Performance and Throughput of Algorithms

Considering the performance and throughput benchmark execution of various encryption algorithms, good results have been achieved. These results demonstrate the performance and throughput of various algorithms that have been conducted in a real and actual manner on a computer, and also show the difference in performance and throughput of a small smart computer with weak computational power (processor), main memory, and cache memory. This experiment was conducted by running a benchmark on a dual-core laptop and a single-core high-end laptop with 2GHz and dual-core. In the second stage, by deactivating one core, the desired benchmark has been executed again. The goal of running a benchmark on a single core is to achieve greater accuracy in the final results.

Considering the benchmark execution on a Pentium 4 computer with a dual-core and 4GB of memory, the performance of the algorithms has varied from each other. In this research, the decryption time and the size of the decrypted message in various algorithms are important. The figure below shows the difference in decryption time on a single core versus two cores for different algorithms. The average decryption of the algorithms in Figure 3 and 4 has been obtained using encryption and decryption of different data sizes.

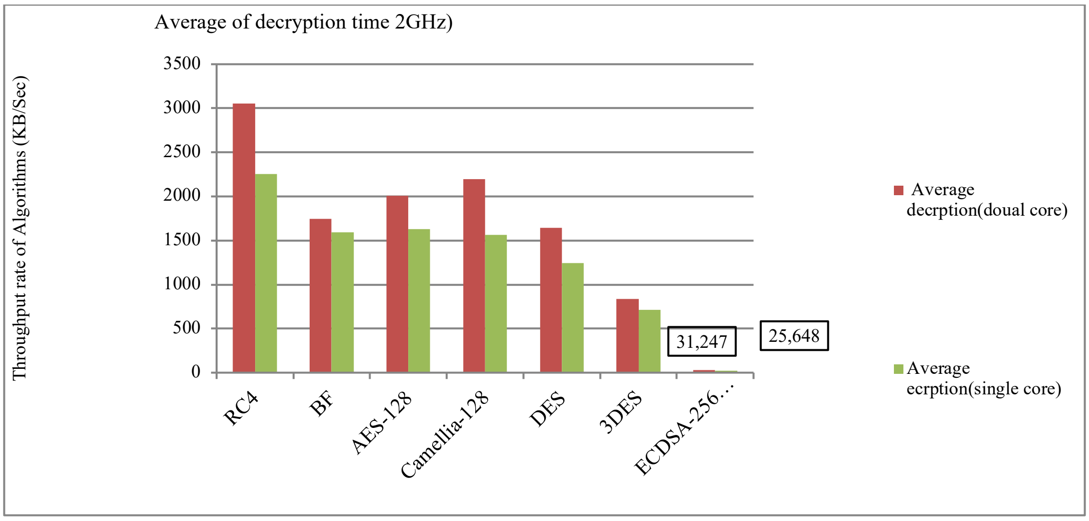

Figure 3: Compare throughput (single-core & dual-core)

The performance results of the algorithms in single-core and dual-core have been obtained after scaling on the 160MHz processor. Additionally, the results and analyses conducted have been based on the average performance of the algorithms. One of the methods used in this experiment is to obtain the decryption (verification) time for each message.

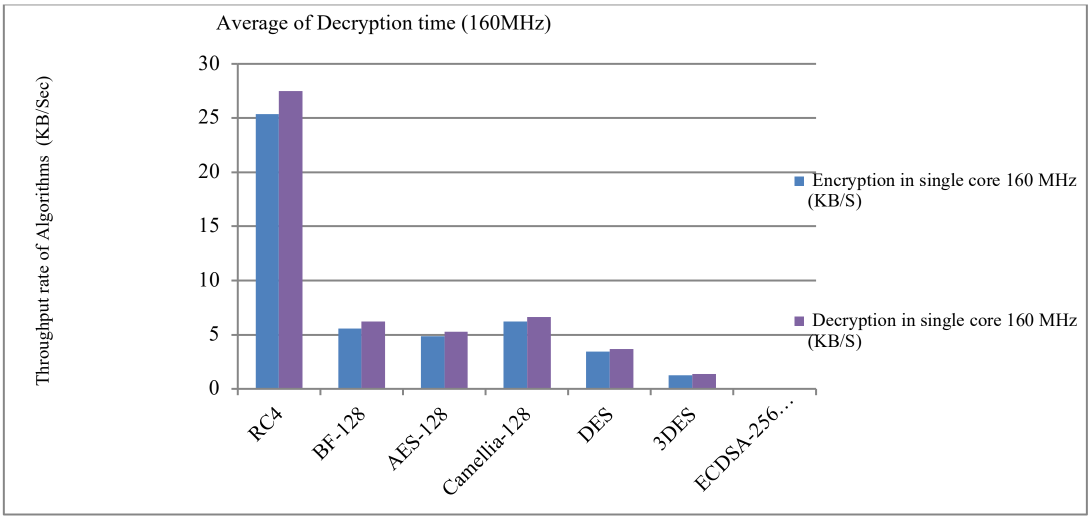

Figure 4: Compare throughput based on benchmark execution on single-core and dual-core

In this method, the message size is specified based on each algorithm; it has been encrypted and decrypted, and the results obtained have been scaled with a 160 MHz processor.

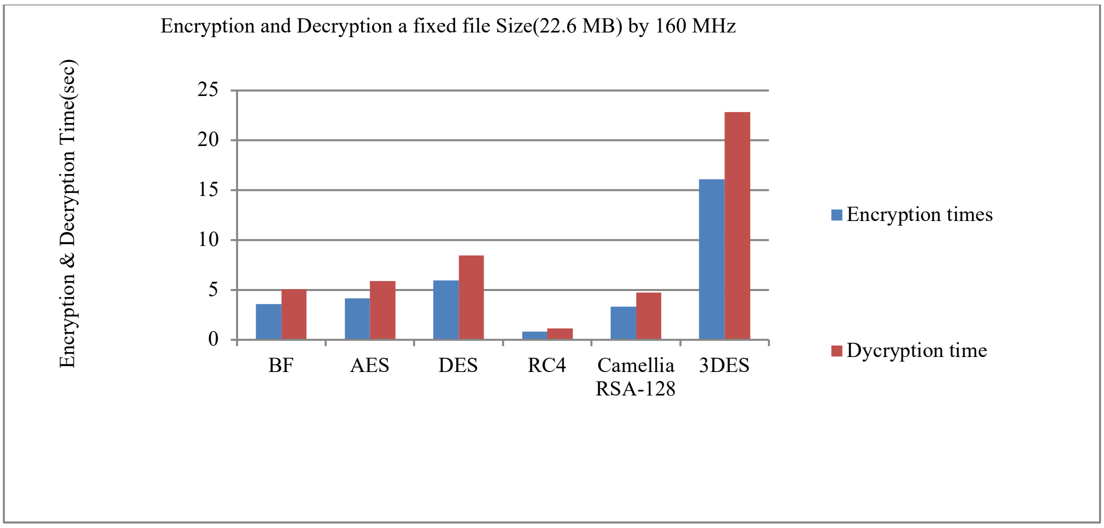

Figure 5: Encryption and Decryption A fixed file size (22.6 MB) on 160 MHZ CPU

Discussion

The results of the present study indicate the performance and efficiency of several cryptographic algorithms while working on an extremely constrained computation device, in this case-a smart meter. Nevertheless, it has several limiting factors that require consideration in the future.

First, experiments were conducted with a limited set of hardware-a 2 GHz processor and, for small devices, a simulated 160 MHz processor. This provides a very useful baseline; however, cryptographic algorithm performance varies widely on different devices, architectures, and in different environmental conditions. For example, other factors not comprehensively looked into included hardware acceleration, size of cache, and multi-core processing. Further studies could be made on a more diverse set of hardware

.

configurations, including real-world smart devices, to better understand the generalizability of these results [1][22].

Second, the current research investigates only a tiny portion of cryptographic algorithms. Although the ones selected are among the most widely used, newer algorithms and protocols may provide different performance profiles [2]. It would be useful for comprehensive experiments on the evaluation of their applicability in resource-constrained environments to include such newer algorithms in future work.

Additionally, most of the executions in the current study have been made with regards to execution time and throughput; other performance metrics such as power consumption, memory usage, and latency are rarely explored. These factors are vital for IoT devices, especially battery-powered ones such as smart meters [3][7]. Energy efficiency and system overheads could be part of the performance evaluation in future studies to give a more holistic understanding of trade-offs in the selection of cryptographic algorithms.

Finally, this study does not take into account network factors that contribute so much to changing the parameters of real cryptographic operations in real-world applications, such as transmission delay and bandwidth limitation. Further research should adopt a more integrated approach, involving assessment of cryptography-network infrastructure-system architecture interaction in IoT systems [4][8].

4. CONCLUSION

This article briefly discusses the advantages and disadvantages of cryptographic algorithms and evaluates them comparatively. This comparison has been conducted based on experimental work in a real-world environment on various encryption algorithms.

In terms of the efficiency and throughput of encryption algorithms, used OpenSSL benchmark for evaluation the resource performance of smart devices. In this experimental are executed many times the encryption and decryption process on dual core and single core CPUs. Every encryption and decryption algorithms are executed more many times. In addition, the comparative tables have been extracted based on the literature review and have been compared and examined from various perspectives. The algorithms that have been compared and examined in this experiment are: AES, DES, 3DES, RC4, Bluefish, RSA, Camellia, and ECDSA. This experience shows that different algorithms vary greatly in terms of speed, efficiency, and throughput. So, the ability of execution encryption algorithms in small smart devices such as smart electricity meters have been achieved. Based on this experience, the throughput and efficiency of smart devices for executing various encryption algorithms have been obtained. The result of this evaluation shows that the RC4 algorithm has the highest throughput and performing encryption at 25KB/Sec and decryption at 27.4KB/Sec. The Camellia algorithm with a 128-bit key length is into second category, the Bluefish the third category, the AES algorithm with a 128-bit key length the fourth category, the DES and 3DES algorithms the fifth category, and the ECDSA algorithm with 128 and 256-bit key lengths into the sixth category, all exhibiting poor operational performance and efficiency. All of these are evaluation are id on single core 160 GHz CPUs with 8KByts RAM which are the smart meter specification. The ECDSA algorithm with a 256-bit key has an encryption speed of 1.37KB/Sec and a decryption speed of 0.0025KB/Sec. It means that small smart devices or smart meters due to resource limitations, are unable to execute advanced security algorithms with large key lengths or operate very slowly.

REFERENCES

[1] Y. Kumar, R. Munjal and H. Sharma, "Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures," *International Journal of Computer Science and Management Studies,* vol. 03, no. 11, pp. 60 -64, 2011.

[2] R. Marqas, S. M. Almufti and R. Rebar Ihsan, "Comparing Symmetric and Asymmetric cryptography in message encryption and decryption by using AES and RSA algorithms," *Journal of Xi'an University of Architecture & Technology,* vol. XII, no. III, p. 3110, 2020.

[3] M. A. Panhwar and S. A. khuhro, Gha, "SACA: A Study of Symmetric and Asymmetric Cryptographic Algorithms," *IJCSNS International Journal of Computer Science and Network Security,* vol. 19, no. 1, pp. 48 - 56, 2019.

[4] www.engage-consulting.co.uk, "High-level Smart Meter Data Traffic Analysis," Engage Consulting Limited, T 0207 4050740, 2010.

[5] M. M. Fouda , Z. M. Fadlull and N. Kato, "A Lightweight Message Authentication Scheme for Smart Grid Communications," *IEEE TRANSACTIONS ON SMART GRID,* vol. 2, no. 4, pp. 675-686, 2011.

[6] W.-K. Yu and S.-F. Wang, "Systematic literature review: comparison study of symmetric key and asymmetric key algorithm," in *2nd Nommensen International Conference on Technology and Engineering IOP Publishing*, Mantreal, Canada, 2018.

[7] V. Gopal, J. Guilford and E. Ozturk, "Improving OpenSSL Performace," White Paper, US. California , 2011.

[8] O. G. Abood and S. K. Guirguis, "Enhancing Performance of Advanced Encryption Standard for Data Security," *International Journal of Engineering and Information Systems (IJEAIS),* vol. 2 , no. 11, pp. 32-38, 2018.

[9] Z. M. Fadlullah, M. M. Fouda and S. Shen, "An Early Warning System Against Malicious Activities for Smart Grid Communications," *IEEE Network Magazine,* vol. 25, no. 5, pp. 1-7, 2011.

[10] K. Alfaheid, *A SECURE AND COMPROMISE-RESILIENT ARCHITECTURE FOR ADVANCED METERING INFRASTRUCTURE,* Oshawa, Ontario, Canada: University of Ontario Institute of Technology (UOIT), 2011.

[11] M. A. Panhwar and S. A. khuhro, "CA: A Study of Symmetric and Asymmetric Cryptographic," *IJCSNS International Journal of Computer Science and Network Security,* vol. 19 , no. 1, 2019.

[12] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions," *J. Ambient Intell. Humaniz. Comput*., pp. 1–18, 2024.

[13] W. Wu, "Application and effectiveness of IoT edge and fog computing technologies in smart energy development with the use of encryption algorithms and security systems," *Comput. Informat*., vol. 43, no. 5, pp. 1029–1052, 2024.

[14] S. Sicari, A. Rizzardi, G. Dini, P. Perazzo, M. La Manna, and A. Coen-Porisini, "Attribute-based encryption and sticky policies for data access control in a smart home scenario: A comparison on networked smart object middleware," *Int. J. Inf. Secur*., vol. 20, pp. 695–713, 2021.

[15] C. Silva, V. A. Cunha, J. P. Barraca, and R. L. Aguiar, "Analysis of the cryptographic algorithms in IoT communications," *Inf. Syst. Front*., vol. 26, no. 4, pp. 1243–1260, 2024.

[16] R. M. A. Al_Azzawi and S. S. M. Al-Dabbagh, "A lightweight encryption algorithm to secure IoT devices," *MINAR Int. J. Appl. Sci. Technol*., vol. 5, no. 3, pp. 37–62, 2023.

[17] H. Zhou, “Comparison of encryption algorithms for wearable devices in IoT systems,” *Eng. Adv*., vol. 3, no. 2, pp. 144–148, 2023.

[18] S. Maitra, D. Richards, A. Abdelgawad, and K. Yelamarthi, "Performance evaluation of IoT encryption algorithms: Memory, timing, and energy," *in 2019 IEEE Sensors Applications Symposium (SAS),* Mar. 2019, pp. 1–6.

[19] A. Kaur and G. Singh, "Encryption algorithms based on security in IoT (Internet of Things)," in 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), Oct. 2021, pp. 482–486.

[20] L. A. Tawalbeh and T. F. Somani, "More secure Internet of Things using robust encryption algorithms against side channel attacks," in 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Nov. 2016, pp. 1–6.

[21] B. W. Aboshosha, M. M. Dessouky, and A. Elsayed, "Energy efficient encryption algorithm for low resources devices," ARCHive-SR, vol. 3, no. 3, pp. 26–37, 2019.

[22] S. Maitra, D. Richards, A. Abdelgawad, and K. Yelamarthi, "Performance evaluation of IoT encryption algorithms: Memory, timing, and energy," in 2019 IEEE Sensors Applications Symposium (SAS), Mar. 2019, pp. 1–6.

.