Article

Not peer-reviewed version

A Comparative Study on the Performance of Security Mechanisms in Internet of Things Devices

Moser José *

Posted Date: 7 June 2023

doi: 10.20944/preprints202306.0529.v1

Keywords: Internet of Things; Performance; Raspberry Pi; Security; Devices; OpenSSL; Hash functions; Tests.

Article

A Comparative Study on the Performance of Security Mechanisms in Internet of Things Devices

Moser José [1,2,*]

[1] Instituto de Telecomunicações and Department of Computer Science, Universidade da Beira Interior, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal; moser.jose@ubi.pt or moser.jose@outlook.com

[2] Department of Informatics, Electronics, and Telecommunications, Polytechnic Institute of Huambo, Angola

Abstract: Data security is a fundamental aspect to be considered in Internet of Things (IoT) information gathering systems, as IoT is a network of interconnected devices that collect and share real-time data, becoming increasingly prevalent in our lives. However, data security in IoT systems presents unique challenges due to the large number of devices and access points involved. This study aims to conduct a literature review on IoT security to analyze the performance of security mechanisms on current development platforms, specifically on a Raspberry Pi 3. Some functions from the OpenSSL library were used, including popular hash functions and cipher algorithms. Additionally, a bash code was developed to obtain the time spent in seconds and the memory consumption in kilobytes. In addition to time and memory calculations, statistical values such as variance and standard deviation were also obtained and compared with results obtained on a personal computer. The tests conducted in this study demonstrated that it is possible to implement these algorithms on platforms with more limited resources, with AES and RSA algorithms being the most suitable for IoT scenarios.

Keywords: internet of things; performance; Raspberry Pi; security; devices; OpenSSL; hash functions; tests

1. Introduction

The Internet of Things (IoT) is a new paradigm [1,2] that enables the interconnection of intelligent physical objects ("things") from our everyday life to the Internet and among themselves, greatly enhancing their utility in the context they operate, especially for humans. The term "Internet of Things" was coined in the late 1990s by Kevin Ashton [3] to identify a collection of sensors and devices connected in intelligent environments that share information with each other and beyond [4,5].

There are several technologies that are commonly associated with IoT due to their specificity, such as Radio Frequency Identification (RFID), Wi-Fi, ZigBee, Sigfox, Low Power Wide Area Network (LoRaWAN), Z-Wave, Bluetooth Low Energy (BLE), IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), and communication protocols like Constrained Application Protocol (CoAP), Message Queuing Telemetry Transport (MQTT), Extensible Messaging and Presence Protocol (XMPP), distributed across the various layers of the IoT architecture [6–13]. The growth of IoT applications results in a large amount of data being generated in various areas of human life, such as healthcare, industries, agriculture, education, commerce, smart cities, smart homes, and more [14,15].

With this in mind, the International Data Corporation (IDC) conducted a study in 2021, which projected that by the year 2025, there would be over 55.7 billion IoT devices in use, generating nearly 80 Zettabytes (ZB) of data [16]. On the other hand, according to Gartner [17], they expect the number of IoT devices to remain around 20 billion by 2025. Therefore, the demand for analyzing large amounts of data generated by IoT devices is increasing. However, such a significant number of internet-connected devices brings forth various challenges and great responsibilities.

IoT is in a very premature state in terms of security and privacy [18]. There is no general approach (or even concern) to security, and there is no standardized mechanism for data and device protection. The enthusiasm surrounding new IoT systems leads to a reduction in time to market, which benefits functionality but hinders security engineering. The less robust processing and memory specifications

 doi:10.20944/preprints202306.0529.v1

for many IoT devices create additional challenges for designing and integrating security mechanisms. Furthermore, the physical accessibility of many devices without strict access control has facilitated the occurrence of vulnerabilities for malicious attacks on IoT devices. The presence of these vulnerabilities creates several challenges for IoT, which are difficult to solve due to a set of constraints [19]. Firstly, the lack of strict physical control over IoT devices increases the chances of a malicious attacker obtaining sensitive user information. Secondly, as most communications are conducted through wireless network technologies, it is possible for attackers to exploit inherent vulnerabilities in these technologies to target IoT devices. Lastly, IoT devices are characterized by limited resources in terms of power, memory, and processing, making it challenging to implement robust security mechanisms.

The present study aimed to conduct a literature review on security in IoT to analyze the performance of security mechanisms on current development platforms, specifically on a Raspberry Pi 3. Additionally, a set of performance tests were conducted on widely used modern cryptography algorithms in computer security solutions, using a specific IoT device, the Raspberry Pi 3.

1.1. Related Works

The literature in the field of IoT, specifically regarding security, has significantly increased in recent years. Therefore, this section focuses solely on the description of related works published after 2016.

The authors [20] propose a security mechanism that classifies user data according to its sensitivity, based on access control, and to enhance security, strong authentication mechanisms are incorporated to ensure the security of the proposed mechanism. The authors analyze security requirements focused on IoT and use these requirements to ensure maximum user privacy.

In [21], a list of security requirements for IoT environments is proposed, which are addressed through an access control mechanism. Additionally, the authors comprehensively address vulnerabilities and security threats in the IoT ecosystem. They state that while it is practically impossible for a single mechanism to guarantee complete security in this environment, the proposed security mechanism can elevate security levels for devices and the overall environment, based on conducted tests and discussions.

The authors in [22] provide a generic analysis of security in IoT, with a focus on the communication technologies RFID and Wireless Sensor Networks (WSN), widely used in this environment. Based on the security requirements for each layer of the IoT architecture, the authors propose a security model, but without specifically creating a secure mechanism for the various security challenges faced by IoT. Instead, the authors focus on existing solutions for network attacks.

The authors in [23] presented a novel security methodology for IoT environments that takes into consideration the security requirements engineering proposed in a work published by [24]. They argue that traditional security approaches used in conventional networks are not suitable for IoT and propose the use of lightweight cryptographic algorithms to ensure security. The methodology is designed to enable the analysis and adaptation of security requirements for IoT at each phase of system development.

In [25], a framework called IoT Hardware Platform Security Advisor (IoT-HarPSecA) was proposed to address the challenge of choosing the right cryptographic algorithms to ensure security in an IoT platform. This framework would assist in selecting specific security algorithms based on specific requirements such as security goals, hardware specifications, message payload size, application area, and power requirements. The tool could help electronics and computer engineers, as well as application developers who are not security experts, make informed decisions about which security algorithms to use in their applications.

The study conducted by the authors [26] presented a review of security standards and evaluation frameworks, including various publications from the National Institute of Standards and Technology (NIST) on security techniques. The aim of the study was to highlight the key areas of focus in existing standards and evaluation frameworks to find solutions that can meet the security needs of IoT devices

 doi:10.20944/preprints202306.0529.v1

and identify the most suitable security techniques and methodologies to ensure the security of IoT devices, through the analysis of different existing standards and frameworks.

In this study [27], a systematic literature review was conducted to analyze the security of IoT devices and propose countermeasures using mobile computing. IoT devices operate in various domains and are exposed to security threats and risks. Therefore, a robust security mechanism is indispensable to address these issues. The innovative approach of this study employed mobile computing infrastructures such as smartphones and applications to tackle the security challenges of IoT. Specific security challenges and problems of IoT were identified, and solutions based on mobile computing hardware and software were proposed. This pioneering research paves the way for future studies on IoT security and emphasizes the importance of integrating mobile computing to ensure the protection of IoT devices.

The authors in [28] discuss the growing research interest in adapting the IoT to industrial domains due to the rapid advancement of technology and the specific topology of industrial IoT. They highlight critical challenges related to security, information preservation, node transactions, communication, trust, privacy, and security protection. These challenges represent limitations and issues for the industrial sector, impacting data integrity, reliability of information exchange, and service delivery prospects. The authors also mention the intersection of blockchain and industrial IoT as a prominent research area but point out the emerging limitation of industrial IoT and connected nodes' inadequate performance, as well as the high resource requirement for authorized private blockchain ledgers. They propose a Hyperledger-based blockchain framework that aims to provide a secure and reliable environment for industrial service execution and transactions. Multiple proof-of-work is investigated and simulated to test the information exchange among connected devices in the industrial IoT, considering resource constraints and ledger storage security.

In this paper [29], the authors discuss the importance of security and privacy in the IoT and the challenges related to authentication in this context. They emphasize that traditional network security cannot be directly applied to IoT networks due to their resource limitations and storage capabilities. The article addresses authentication mechanisms in the IoT, highlighting attacks and technical methods in this domain. Existing security verification techniques, authentication evaluation schemes, and analysis of current protocols are also discussed. The objective of the study is to provide relevant information for future researchers, addressing security issues, open challenges, and future perspectives in IoT authentication.

In the text, the authors [30] discuss the connectivity of the IoT, highlighting the diversity of devices that can be connected, such as smartphones, coffee makers, washing machines, cars, lamps, and wearable devices. They emphasize that the rapid growth of connected sensors and devices bridges the gap between the physical and digital world, bringing benefits to individuals, processes, and businesses. However, security is a significant challenge for most of these applications, as the lack of secure links exposes the exchanged data between devices to theft and attacks, generating interest from hackers. Secure communication in the IoT requires a multifaceted approach, including communication protocols and data protection. An important aspect is the initialization of keys in devices to support secure communications. The article examines key bootstrapping protocols based on public-key cryptography in the IoT, which are relevant to the implementation of distributed identity and trust management mechanisms. The proposals are analyzed and classified based on key delivery methods, underlying cryptographic primitives, and supported authentication mechanisms. The authors also identify and discuss the main challenges in implementing these methods in IoT applications and devices.

This article [31] redefines the IoT ecosystem based on key technologies and proposes a modified three-layer IoT architecture, where the perception layer is divided into elementary blocks with assigned functions. Enabling technologies, attacks, and security countermeasures are classified in each layer of the proposed architecture. The role of emerging technologies in IoT security is discussed, presenting the security aspects of prominent standards and recent technologies that can

 doi:10.20944/preprints202306.0529.v1

influence the evolving architecture of IoT. Special attention is given to Intelligent RF Connectivity (IQRF) technology, which provides packet-oriented wireless communication in the sub-GHz band (868 MHz). The security aspects implemented in this technology are highlighted and compared with other established technologies. Lightweight security solutions are also presented to mitigate threats within the proposed architecture for the IoT ecosystem

In the article [32], a robust domain distributed trust management system called RobustTrust is proposed, which allows a device to locally evaluate trust in relation to other devices. This system divides trust into three security components that help IoT nodes become resilient against compromised and malicious devices and nodes. The proposed mechanism introduces innovations such as high scalability, multiple evaluation components to enhance resilience against attacks, and the use of recommendations and feedback to build knowledge. Additionally, the mechanism is event-driven, which contributes to a more effective trust evaluation and increases system efficiency. The proposed work is compared with other available trust evaluation schemes, considering attributes such as reliability, usability, accuracy, among others. The RobustTrust system is validated through extensive simulations, taking into account the performance of absolute trust value, trust estimation accuracy, and various potential attacks.

This article [33] provides a comprehensive review of different network anomaly mitigation schemes in IoT networks. The objectives, operational procedures, and strengths of each scheme are discussed. Additionally, a comparison table of the reviewed schemes, along with a taxonomy based on the detection methodology employed, is presented. This study offers both qualitative and quantitative evaluations. Performance assessments of selected classification algorithms used in IoT network anomaly mitigation schemes were conducted using the UNSW-NB15 dataset. Furthermore, the challenges and open issues in the development of these mitigation schemes are discussed. The work emphasizes the ongoing importance of enhancing security in evolving IoT networks to ensure device protection and user information confidentiality.

This article [34] describes a hardware security project with an enhanced architecture for detecting buffer overflow attacks in IoT devices. The project includes instruction monitoring and verification to track program execution behavior, as well as secure tag validation to monitor the attributes of each memory segment. During compilation, automated extraction tools extract the monitoring model and secure tags from each memory segment. During runtime, the hardware observes the dynamic execution tracing and verifies if it complies with the allowed behavior. If not, appropriate response mechanisms are triggered. The proposed schemes do not require changes to the compiler or existing instruction set and do not impose restrictions on software developers. The architectural design has been implemented on a real OR1200-FPGA platform. Experimental analysis demonstrates that the proposed techniques can detect a wide range of buffer overflow attacks with low performance overhead and minimal overhead expenses.

This article [35] addresses the increasing demand for quality services and infrastructure in smart societies, specifically in the context of the Industrial Internet of Things (IIoT). One of the challenges faced in smart urbanization is the Secure Management of Energy Demand (DSM). IIoT exposes industrial systems to vulnerabilities such as malware, cyber attacks, and security risks. To tackle these challenges, the article proposes a secure and reliable multi-layered DSM mechanism using IIoT-based big data analytics. The main objective is to provide a generic and secure solution for smart societies in the IIoT environment. The proposed mechanism adopts a centralized approach to achieve optimal DSM in a home area network. To enhance security, a payload-based authentication scheme is used, relying on a lightweight handshake mechanism. The proposed method leverages the lightweight resources of the constrained application protocol to enable clients to efficiently monitor multiple resources on the server in terms of energy consumption. Additionally, data streams are processed using big data analytics with MapReduce parallel processing. The proposed authentication approach is evaluated using NetDuino Plus 2 boards, demonstrating lower connection overhead, memory consumption, and response time, while providing robust defense against various malicious attacks.

 doi:10.20944/preprints202306.0529.v1

On the other hand, the data processing approach is tested on reliable datasets using Apache Hadoop with Apache Spark to validate the proposed DSM mechanism.

1.2. Comparison of related works

Many studies emphasize the complexity of developing a comprehensive security solution for the IoT. It is concluded, from the concise analysis conducted in the previous section, that there is no single approach that can guarantee overall IoT security. Instead, a combination of controls and security mechanisms working in harmony is necessary to address the various challenges in this diverse context. Security in IoT systems is a critical concern as it involves a variety of interconnected devices, sensors, networks, and applications. Protecting personal data, ensuring information confidentiality, and maintaining device integrity are just a few of the fundamental concerns in this complex ecosystem. Given the complexity and ongoing evolution of IoT, it is essential to continue researching and developing innovative security solutions. Only through a multidimensional and collaborative approach involving security experts, engineers, policymakers, and end-users can we advance the protection of systems and ensure user trust and privacy in this increasingly connected world.

To gain a better understanding of the current landscape of IoT security, a comparison of related works is presented in Table 1. The table covers three main dimensions: firstly, whether the works define and specifically focus on IoT as the subject of study; secondly, whether they explore specific security mechanisms such as authentication, encryption, or access control; and finally, whether they include security testing to assess the effectiveness and robustness of the proposed systems.

In the Table 1, the symbols $\star$, $\diamond$ and $\times$ are used to indicate whether a particular aspect is detailed in the article, mentioned superficially, or not covered in the article, respectively. This comparative analysis provides a clearer view of the existing approaches and the gaps that still need to be filled in the field of security in the IoT.

Table 1. Comparison of related works.

Authors and Citation	Year	IoT Definition	Security Mechanisms	Security Tests
Kaliya at al [20]	2017	$\star$	$\star$	$\diamond$
Pal at al [21]	2017	$\star$	$\star$	$\times$
Daud at al [22]	2017	$\star$	$\star$	$\times$
Josyula at al [23]	2017	$\star$	$\star$	$\diamond$
Samaila at al [25]	2019	$\star$	$\star$	$\star$
Wahab at al [26]	2021	$\star$	$\star$	$\star$
Liao at al [27]	2020	$\star$	$\star$	$\diamond$
Ayub at al [28]	2022	$\star$	$\diamond$	$\times$
Nandy at al [29]	2019	$\star$	$\diamond$	$\times$
Malik at al [30]	2019	$\star$	$\star$	$\star$
Bouzid at al [31]	2022	$\star$	$\star$	$\star$
Awan at al [32]	2019	$\star$	$\star$	$\star$
Lawal at al [33]	2020	$\star$	$\star$	$\star$
Xu at al [34]	2018	$\star$	$\star$	$\diamond$
Babar at al [35]	2018	$\star$	$\star$	$\diamond$

2. Materials and Methods

2.1. Algorithms

The algorithms mentioned in this study are commonly used in the development of applications for both web and IoT devices. However, it is important to note that some of these algorithms are not easily adaptable to devices with memory, processing, and energy constraints. Therefore, it is relevant to perform a comparative analysis of their performance on typical IoT platforms and conventional computers. There are various types of cryptographic algorithms that are considered for performance

 doi:10.20944/preprints202306.0529.v1

testing. The present study focuses on analyzing the following cryptographic algorithms: Symmetric Key algorithms and Public Key algorithms.

2.1.1. Symmetric Key algorithms

The cryptographic algorithms of symmetric key, Data Encryption Standard (DES), and Triple Data Encryption Algorithm (3DES) are fixed-block size cipher algorithms [36]. DES, with a 56-bit key (7 bytes) and a 64-bit block (8 bytes), is currently considered insecure but holds historical significance as the first internationally standardized symmetric key algorithm. On the other hand, 3DES combines the encrypt-decrypt-encrypt operations of DES with three keys to enhance its cryptographic strength. Both ciphers operate with rounds and utilize Feistel networks in their encryption and decryption operations. In the conducted tests, the ciphers were operated in the Electronic Codebook (ECB), Cipher Block Chaining (CBC), Output Feedback (OFB), and Cipher Feedback (CFB) modes. Despite the recommendation against using DES, it was included in the tests due to its historical importance.

The Advanced Encryption Standard (AES) is a block cipher symmetric key algorithm that has become the international standard for symmetric key encryption, replacing DES. AES is capable of using three different key sizes, which provide different levels of security: 128, 192, and 256 bits. The block size on which AES operates is always fixed at 128 bits [37]. Additionally, AES can operate in various cipher modes, such as ECB, CBC, Counter (CTR), CFB, and OFB. All of these modes were tested in the context of this study. When used correctly, AES is considered highly secure and efficient in terms of performance, making it an ideal choice for symmetric key encryption in IoT devices and systems.

The Rivest Cipher 4 (RC4) is a stream cipher algorithm, also known as a symmetric key stream cipher. Developed by Ron Rivest in 1987, initially as a cryptographically secure pseudo-random number generator, RC4 has been widely used in various security environments and protocols [38]. Despite the identification of weak keys and vulnerabilities, RC4 is still used in some systems, mainly due to its low computational overhead. The algorithm uses keys ranging from 40 to 128 bits and has been widely used in security standards such as Transport Layer Security (TLS), Wired Equivalent Privacy (WEP), and Wi-Fi Protected Access (WPA) until 2015 [37]. Although considered in the context of this study due to its historical significance, it was also used for comparative purposes in relation to the other algorithms discussed in this paper.

2.1.2. Public Key algorithms

The Rivest Shamir Adleman (RSA) is a public key cryptographic algorithm that is widely used for encrypting and decrypting small amounts of data, such as bit strings smaller than one of the parameters of the public key, known as the modulus. Created by Ronald Rivest, Adi Shamir, and Leonard Adleman in 1978, it is considered one of the greatest advancements in public key cryptography [39]. RSA is ideal for encrypting and exchanging cryptographic secrets and symmetric cipher keys, as well as for signing hash values smaller than the modulus. Based on Number Theory, RSA relies on the complexity of the mathematical problem of factoring an extremely large composite number into primes to ensure its security [26]. The encryption and decryption operations involve modular exponentiation, making the implementation relatively simple and easy to understand. The RSA also utilizes trapdoor one-way functions, which cannot be efficiently reversed unless the private key is known. This public key approach is widely used to ensure the privacy and security of sensitive data in various applications, including email message encryption, user authentication in online systems, and security of financial transactions.

To encrypt a message $M \epsilon Z_n$, the public key pk consisting of two numbers (N, e) is used, and the calculation $C \leftarrow M^e (mod\ N)$ is performed [40,41]. To decrypt the ciphertext, the private key sk is used in the calculation $M \leftarrow C^d (mod\ N)$, where M is the message, C is the ciphertext, and e and d are integers. N is the result of multiplying two large prime numbers $(\geqslant 2^{1024})$. It should also be noted that

 doi:10.20944/preprints202306.0529.v1

the key sizes mentioned above were used in this study to sign and verify files with sizes of $100MB$, $1GB$, and $2GB$ using the Secure Hash Algorithm 256 (SHA256) [36].

2.2. Methods

All the encryption algorithms mentioned in the subsection 2.1 were used for performance testing, and in doing so, we were able to analyze certain distinct behaviors. These tests allowed us to identify the encryption algorithms that best adapt to the world of IoT (e.g., devices and systems) with resource levels similar to our test device, based on the efficiency of their encryption and decryption methods. The tests involved measuring the time taken in seconds and the memory consumption in kilobytes. In addition to the calculations related to time and memory, statistical values such as variance and standard deviation were also obtained.

To perform the tests, the implementations of the algorithms defined in the Openssl library, version 1.1.0.2g, were used, with a fixed encryption key and initialization vector (when applicable). For measuring time and memory consumption, the *time tool* was used with the e and m parameters, respectively. The variance $var(x)$ and standard deviation σ were obtained using the equations:

$$var(x) = \sum \frac{(x_i - \bar{x})^2}{n-1} \tag{1}$$

$$\sigma = \sqrt{\sum \frac{(x_i - \bar{x})^2}{n-1}} \tag{2}$$

The equation (1) and equation (2) represent the calculation of sample variance and sample standard deviation, respectively. In these equations, $\sum$ denotes summation, $(x_i - \bar{x})$ represents the difference between each value x_i and the mean of the values x, $\bar{x}$. n represents the number of elements in the sample, and $(n\text{-}1)$ is the Bessel's correction factor used to estimate the population variance $var(x)$ and the population standard deviation σ of a variable x from the sample.

The tests were conducted on a Raspberry Pi 3 (Cortex A53 Quad Core, ARM Cortex, 1.2 GHz, 16GB storage, 1GB memory) with the Ubuntu MATE 16.04.2 operating system, and on an ACER Aspire ES15 computer (AMD Quad-Core, A5-5000 1.5 GHz, 1GB HDD, 4GB DDR3 memory) with the Ubuntu 18.04 operating system. A total of 100 repetitions were performed for each file size of 100MB, 1GB, and 2GB, both on the Raspberry Pi 3 and the computer.

3. Results

This section presents the results obtained from the tests conducted on the subject matter. It is important to note that the results obtained from the DES-CBC, 3DES-CBC, AES-CBC, RC4 ciphers, as well as the results obtained from the RSA signature and verification operations, are compared with the results obtained on a personal computer. This comparison allows for evaluating and contrasting the efficiency and performance of the studied algorithms and operations in different environments. The test results provide an objective basis for understanding the performance differences between the encryption algorithms and the signature and verification operations using RSA.

By comparing them with a personal computer, one can gain insights into the impact that specific characteristics of the Raspberry Pi 3, such as its limited processor and memory, can have on the performance of these algorithms and operations. This comparative analysis can assist in selecting the most suitable algorithm for use in IoT devices with limited resources, such as the Raspberry Pi 3.

The results of the performance analysis of the encryption algorithms DES, 3DES, RC4, and AES were presented in Tables 2 and 3, showcasing the time required to encrypt and decrypt files of 100MB and 1GB on a Raspberry Pi 3 and a computer. It is evident that the AES algorithm proved to be the most efficient on devices with reasonable computational power, such as the utilized Raspberry Pi 3, for both smaller and larger files. On the other hand, the 3DES cipher exhibited significantly inferior performance, while the RC4 cipher approached the performance of AES for larger files. However, as

 doi:10.20944/preprints202306.0529.v1

emphasized in the previous section, AES offers superior security, making it the best choice among the analyzed algorithms.

Table 2. Results regarding the processing time of the DES-CBC, 3DES-CBC, RC4, AES-CBC, and RSA algorithms on files of 100MB and 1GB for the Raspberry Pi 3.

Files	Algorithms		Encrypt/Sign			Decrypt/Verify		
	Cipher/mode	Key(bits)	*Time(s)*	$var(x)$	σ	*Time(s)*	$var(x)$	σ
100*MB*	DES-CBC	56	9,86	105	10,25	10,4	107	10,34
	3DES-CBC	168	16,37	275,29	16,59	17,84	324,52	18,01
	RC4	128	8,44	86,40	9,30	8,75	90,27	9,50
	AES-CBC	128	7,89	71,44	8,45	8,80	89,23	9,45
		192	7,46	73,33	8,56	7,90	72,08	8,49
		256	7,46	63,70	7,98	7,78	69,15	8,32
	RSA	2048	1,46	2,17	1,47	1,44	2,07	1,44
		4096	1,58	2,52	1,59	1,44	2,09	1,44
1*GB*	DES-CBC	56	137,98	19236,86	192,37	138,39	19156,98	191,57
	3DES-CBC	168	216,02	47135,37	217,11	220,46	48606,91	486,07
	RC4	128	129,91	17053,43	130,59	129,80	16855,74	168,56
	AES-CBC	128	129,98	17070,49	170,70	129,33	16732,45	167,32
		192	130,72	17269,09	172,69	130,58	17060,29	170,60
		256	131,72	17365,94	173,66	130,59	17057,44	170,57
	RSA	2048	46,44	2178,68	46,68	46,42	2154,67	46,71
		4096	46,86	2217,80	47,09	46,71	2181,94	46,71

Table 3. Results regarding the processing time of the DES-CBC, 3DES-CBC, RC4, AES-CBC, and RSA algorithms on files of 100MB and 1GB for the Personal Computer.

Files	Algorithms		Encrypt/Sign			Decrypt/Verify		
	Cipher/mode	Key(bits)	*Time(s)*	$var(x)$	σ	*Time(s)*	$var(x)$	σ
100*MB*	DES-CBC	56	4,56	20,97	4,58	4,48	20,07	4,48
	3DES-CBC	168	11,50	133,49	11,55	11,56	133,58	11,56
	RC4	128	0,98	0,98	0,99	0,96	0,94	0,97
	AES-CBC	128	1,14	1,35	1,16	0,87	0,78	0,89
		192	1,14	1,33	1,16	0,87	0,78	0,88
		256	1,18	1,43	1,20	0,82	0,68	0,82
	RSA	2048	1,29	167	1,29	1,28	1,63	1,28
		4096	1,32	1,76	1,33	1,28	1,64	1,28
1*GB*	DES-CBC	56	59,20	3549,42	59,58	76,69	3466,42	58,88
	3DES-CBC	168	128,58	16702,56	129,24	116,45	18394,70	135,63
	RC4	128	31,68	1016,74	31,89	25,12	633,48	25,17
	AES-CBC	128	30,45	940,25	30,66	50,47	651,85	25,13
		192	30,01	910,43	30,17	48,05	535,09	23,13
		256	33,76	1156	34	58,30	1160,95	34,07
	RSA	2048	13	170,74	13,07	12,98	168,56	12,98
		4096	13,03	171,55	13,10	12,99	168,70	12,99

 doi:10.20944/preprints202306.0529.v1

Regarding the key sizes used, no significant differences were observed (the largest variation is around 12%), indicating that there is no benefit in using smaller keys. In the case of the RSA signature algorithm, the obtained results on both devices align with expectations, demonstrating a performance approximately three times faster than AES in its encryption process. When comparing the two devices, the computer, as expected, outperformed the Raspberry Pi 3 in both operations, executing the same tasks in approximately 15 to 25% of the time required.

These results underscore the importance of choosing the appropriate algorithm based on specific security and efficiency requirements for different devices and file sizes. Additionally, they highlight the significant influence of the device's computational power on the speed of cryptographic operations.

The Tables 4 and 5 present the results regarding the memory consumption of the DES, 3DES, RC4, and AES algorithms for encrypting and decrypting 100MB and 1GB files on the two devices used. It can be observed that, regardless of the algorithm used, the Raspberry Pi 3 consumes about 60% of the memory used by the computer, with values around 2500 KB. This low memory consumption remains consistent regardless of the file size, indicating that this metric has no significant influence. The use of keys of different sizes does not affect the obtained results, as they are similar among the different algorithms.

Table 4. Results regarding the memory consumption of the DES-CBC, 3DES-CBC, and RSA algorithms on files of 100MB and 1GB for the Raspberry Pi 3.

Files	Algorithms		Encrypt/Sign			Decrypt/Verify		
	Cipher/mode	Key(bits)	*Memory(Kb)*	$var(x)$	σ	*Memory(Kb)*	$var(x)$	σ
100*MB*	DES-CBC	56	2583,16	3984,26	63,12	2585,80	4033,40	63,51
	3DES-CBC	168	2583,32	3575,94	59,80	2586,84	4183,77	64,68
	RC4	128	2577,28	2789,78	52,82	2582,88	2960,03	54,41
	AES-CBC	128	2597,04	3464,93	58,86	2592,24	3214,02	56,69
		192	2601,68	3138,00	56,02	2596,28	3287,76	57,34
		256	2596,12	2663,58	51,61	2598,96	2812,04	53,03
	RSA	2048	2623,56	8223,32	90,68	2579,52	7063,77	84,05
		4096	2629,68	8868,26	94,17	2571,16	7632,73	87,37
1*GB*	DES-CBC	56	2598,24	3993,96	63,20	2589,16	3982,17	63,10
	3DES-CBC	168	2565,84	4272,30	65,36	2568,68	4166,42	64,55
	RC4	128	2570	2681,37	51,78	2572,32	2446,62	249,46
	AES-CBC	128	2598,36	2883,59	53,70	2595,32	2411,86	49,11
		192	2603,92	3522,90	59,35	2594,68	3367,06	58,03
		256	2595,48	2717,63	52,13	2598,80	3575,52	59,80
	RSA	2048	2615,56	7823,48	88,45	2565,88	7833,43	88,51
		4096	2639,00	7436,73	86,24	2562,48	6869,37	82,88

These results demonstrate that the Raspberry Pi 3 requires considerably less memory compared to the computer to perform the encryption and decryption processes using the DES, 3DES, RC4, and AES algorithms. The memory consumption remains constant regardless of the file size, suggesting that the Raspberry Pi 3 operates efficiently within its memory limitations. The utilization of different key sizes does not have a substantial impact on the memory consumption of these algorithms.

It is worth noting that memory consumption is an important aspect to consider when deploying cryptographic algorithms, especially on devices with limited resources like the Raspberry Pi 3. These findings highlight the suitability of the Raspberry Pi 3 for cryptographic operations with the analyzed

 doi:10.20944/preprints202306.0529.v1

algorithms, as it demonstrates efficient memory usage while achieving the desired encryption and decryption functionalities.

However, it is essential to keep in mind that AES is not the only available option. The RSA algorithm, for example, is widely used for digital signature and verification operations, particularly suitable for scenarios that require authentication and data integrity. Although RSA is known for its relatively slower performance compared to AES, its security and ability to establish robust trust are factors that make its use indispensable in many cases. Therefore, when deciding between using AES or RSA, it is important to consider the specific requirements of the application, such as the need for confidentiality, authenticity, and data integrity. In some cases, it may be necessary to use a combination of cryptographic algorithms to meet different security and performance requirements.

Table 5. Results regarding the memory consumption of the DES-CBC, 3DES-CBC, and RSA algorithms on files of 100MB and 1GB for the Personal Computer.

Files	Algorithms		Encrypt/Sign			Decrypt/Verify		
	Cipher/mode	Key(bits)	*Memory(Kb)*	$var(x)$	σ	*Memory(Kb)*	$var(x)$	σ
$100MB$	DES-CBC	56	4452,36	3537,16	59,47	4455,28	5259,32	72,52
	3DES-CBC	168	4489,28	5938,06	77,06	4481,60	5375,36	73,32
	RC4	128	4473,36	5663,10	75,25	4470	7093,28	84,22
	AES-CBC	128	4455,56	4350,03	65,95	4457,28	4785,56	69,18
		192	4461,56	5176,53	71,95	4464,92	4094,03	63,98
		256	4472,48	3997,18	63,22	4465,96	4286,56	65,47
	RSA	2048	4485,88	3060,83	55,32	4546,36	4614,11	67,93
		4096	4524,12	3204,35	56,61	4548,92	4832,59	69,52
$1GB$	DES-CBC	56	4358,96	10885,3	104,33	4369,40	11395,9	106,7
	3DES-CBC	168	4345,44	5169,30	71,90	4322,28	4717,84	68,69
	RC4	128	4466,48	4957,67	70,41	4457,64	5394,91	73,45
	AES-CBC	128	4301,44	3944,25	39,44	4301,44	5140,65	71,70
		192	4308,04	5067,15	71,18	4318,60	4447,48	66,69
		256	4318,08	5340,60	73,08	4317,32	4632,66	68,06
	RSA	2048	4347,20	3132,12	55,97	4406,80	2278,88	47,74
		4096	4366,04	2199,43	46,90	4413,40	1406,84	37,51

4. Discussion

Several discussions have emerged from the results obtained in the present study and have been addressed in the following texts.

The studies conducted in [25,26,30–33] emphasize the crucial importance of implementing robust mechanisms to ensure the security of information in the IoT context. These research studies highlight the increasing interconnectivity of IoT devices and the consequent expansion of attack surfaces, making information protection an increasingly complex challenge. They also emphasize the need to create additional security mechanisms in this environment to reinforce both internal and external security within this network. The findings from these research studies reveal the vulnerability of IoT devices and the opportunity for further studies in the field.

The results of these studies raise awareness about the fragility of IoT devices and underscore the importance of conducting further research in this area to enhance existing security mechanisms and develop new solutions capable of strengthening both internal and external security within this network.

 doi:10.20944/preprints202306.0529.v1

The outcomes of these studies serve as a warning to the vulnerability of IoT devices, which are often designed with limited resources and insufficient security measures. These shortcomings allow attackers to exploit vulnerabilities and gain access to sensitive information, compromising privacy, data integrity, and service availability. Therefore, it is essential to adopt a comprehensive approach to strengthen security in the IoT environment. This includes implementing robust authentication to ensure that only authorized devices and users have access to the network. Additionally, advanced encryption techniques must be used to protect the confidentiality of transmitted and stored data in IoT devices.

One relevant discussion was the trade-off between speed and security. While AES proved to be faster in terms of encryption and decryption, RSA stood out in terms of security. This raised questions about finding the ideal balance between the need for fast processing and ensuring a high level of protection for the data transmitted and stored in IoT devices.

Despite this slight difference, both algorithms are considered secure; however, their security characteristics differ. It is important to analyze and compare known vulnerabilities and theoretical attacks against AES and RSA, as well as the suitability of these algorithms for protecting data in IoT environments. The tests in the study also demonstrated that it is possible to implement these algorithms on platforms with more limited resources.

Another point of discussion was the efficiency in the use of resources in the Raspberry Pi 3. It was observed that AES required fewer processing and memory resources compared to RSA. This efficiency is important in devices that typically have limited resources. However, there was also debate about whether the resource-saving aspect of AES could be significant enough to outweigh the security advantages offered by RSA in certain scenarios.

Although the study focused on the AES, RSA, DES, 3DES, and RC4 algorithms, discussions arose regarding the inclusion of other encryption algorithms in the comparative study. Algorithms such as Blowfish, Twofish, and Elliptic Curve Cryptography (ECC) were mentioned as alternatives that could be considered to evaluate performance and security in IoT devices. Therefore, it is recommended to explore other options for security mechanisms and consider factors such as scalability, ease of implementation, and compatibility with other devices and protocols used in the IoT infrastructure.

Furthermore, there were discussions about the applicability of the results in different contexts and scenarios of IoT. Considering that this ecosystem is complex and diverse, each application may have specific security and performance requirements. Therefore, generalizing the results to all IoT cases was subject to analysis and debate, highlighting the importance of a customized and context-adapted approach.

The discussions generated from the present study reinforced the importance of carefully considering security, performance, and resource efficiency requirements when selecting security mechanisms for IoT devices. Decisions should be based on a comprehensive analysis of the context, taking into account the specific characteristics of each application and the importance of speed, security, and efficiency. On the other hand, choosing the right security mechanisms, considering performance, resource efficiency, and the specific needs of each application, is essential to ensure adequate data protection and system integrity in these devices.

These contributions are just part of the long road to a more secure IoT by design. We believe that studying and reporting the behavior of algorithms and security mechanisms in IoT devices is crucial to better understand their issues and limitations, as well as to think about IoT security in the future and assist in the choice of mechanisms to be implemented.

However, IoT devices are highly vulnerable and often lack security mechanisms and practices that fit the environment. Given the significant increase in IoT adoption, it is extremely important to develop specific techniques and technologies to ensure the security of these devices. This is necessary not only because of the unique environment in which IoT applications operate but also because it is not sufficient to simply transfer existing mechanisms in current networks or systems to the IoT.

 doi:10.20944/preprints202306.0529.v1

Finally, it is necessary to emphasize that continuous research and development are crucial in the area of IoT security. As threats evolve and new vulnerabilities are discovered, it is essential to keep up with advances in cryptography and security and to adopt updated practices and protocols. IoT security is an ever-changing challenge, and it is essential to stay updated and adapt to new requirements and emerging threats.

5. Conclusions

Based on the comparative study conducted on the performance of security mechanisms in IoT devices, we have reached the following conclusions:

We have identified that data security effectiveness is a crucial factor in evaluating security mechanisms.

The AES and RSA algorithms demonstrated a high level of responsiveness in the conducted tests and are more robust and reliable for protecting IoT devices.

It has become evident that additional security measures and mechanisms are necessary to ensure that the IoT continues to advance and achieve its objectives, considering especially the relationship between performance, security, and the specific needs of devices in this context.

References

1. Aman, A.H.M.; Yadegaridehkordi, E.; Attarbashi, Z.S.; Hassan, R.; Park, Y.J. A Survey on Trend and Classification of Internet of Things Reviews. *IEEE Access* **2020**, *8*, 111763–111782. https://doi.org/10.1109/ACCESS.2020.3002932.
2. de Matos, E.; Tiburski, R.T.; Moratelli, C.R.; Filho, S.J.; Amaral, L.A.; Ramachandran, G.; Krishnamachari, B.; Hessel, F. Context information sharing for the Internet of Things: A survey. *Computer Networks* **2020**, *166*. https://doi.org/10.1016/J.COMNET.2019.106988.
3. Ashton, K.; et al. That 'internet of things' thing. *RFID journal* **2009**, *22*, 97–114.
4. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials* **2015**, *17*, 2347–2376. https://doi.org/10.1109/COMST.2015.2444095.
5. Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the Internet of Things: perspectives and challenges. *Wireless Networks* **2014**, *20*, 2481–2501. https://doi.org/10.1007/s11276-014-0761-7.
6. Kamesh.; Priya, N.S. Security enhancement of authenticated RFID generation. *International Journal of Applied Engineering Research* **2014**, *9*, 5968–5974. https://doi.org/10.1002/sec.
7. Amjad, A.; Azam, F.; Anwar, M.W.; Butt, W.H. A Systematic Review on the Data Interoperability of Application Layer Protocols in Industrial IoT. *IEEE Access* **2021**, *9*, 96528–96545. https://doi.org/10.1109/ACCESS.2021.3094763.
8. Babun, L.; Denney, K.; Celik, Z.B.; McDaniel, P.; Uluagac, A.S. A survey on IoT platforms: Communication, security, and privacy perspectives. *Computer Networks* **2021**, *192*. https://doi.org/10.1016/J.COMNET.2021.108040.
9. Kassab, W.; Darabkh, K.A. A–Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations. *Journal of Network and Computer Applications* **2020**, *163*. https://doi.org/10.1016/J.JNCA.2020.102663.
10. Iglesias-Urkia, M.; Orive, A.; Urbieta, A.; Casado-Mansilla, D. Analysis of CoAP implementations for industrial Internet of Things: a survey. *Journal of Ambient Intelligence and Humanized Computing* **2019**, *10*, 2505–2518. https://doi.org/10.1007/S12652-018-0729-Z.
11. Liu, X.; Zhang, T.; Hu, N.; Zhang, P.; Zhang, Y. The method of Internet of Things access and network communication based on MQTT. *Computer Communications* **2020**, *153*, 169–176. https://doi.org/10.1016/J.COMCOM.2020.01.044.
12. Ferrera, E.; Conzon, D.; Brizzi, P.; Rossini, R.; Pastrone, C.; Jentsch, M.; Kool, P.; Kamienski, C.; Sadok, D. XMPP-based infrastructure for IoT network management and rapid services and applications development. *Annales des Telecommunications/Annals of Telecommunications* **2017**, *72*, 443–457. https://doi.org/10.1007/S12243-017-0586-3.

 doi:10.20944/preprints202306.0529.v1

13. Bellavista, P.; Zanni, A. Scalability of kura-extended gateways via MQTT-CoAP integration and hierarchical optimizations. *BodyNets International Conference on Body Area Networks* **2017**. https://doi.org/10.4108/EAI.15-12-2016.2267595.
14. Barros, V.A.; Sérgio, A.B.; Bruschi, S.M.; Monaco, F.J.; Estrella, J.C. An IoT multi-protocol strategy for the interoperability of distinct communication protocols applied to web of things. *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web, WebMedia 2019* **2019**, pp. 81–88. https://doi.org/10.1145/3323503.3349546.
15. Petrova-Antonova, D.; Andreev, G.; Ilieva, S. Unified connectivity of IoT devices through abstraction of application protocols. *ACM International Conference Proceeding Series* **2017**, *Part F131202*, 56–61. https://doi.org/10.1145/3134383.3134385.
16. Hojlo, J. Future of Industry Ecosystems: Shared Data and Insights, 2021.
17. Sasaki, Y. A Survey on IoT Big Data Analytic Systems: Current and Future. *IEEE Internet of Things Journal* **2022**, *9*, 1024–1036. https://doi.org/10.1109/JIOT.2021.3131724.
18. Samaila, M.; Neto, M.; Fernandes, D.A.B.; Freire, M.M.; Inácio, P.R.M. "Challenges of Securing Internet of Things Devices: A Survey". *Wiley Security and Privacy (SPY)* **2017**, *1*, 20. https://doi.org/10.1002/spy2.20.
19. Keoh, S.L.; Kumar, S.S.; Tschofenig, H. Securing the internet of things: A standardization perspective. *IEEE Internet of Things Journal* **2014**, *1*, 265–275. https://doi.org/10.1109/JIOT.2014.2323395.
20. Kaliya, N.; Hussain, M. Framework for privacy preservation in iot through classification and access control mechanisms. 4 2017, pp. 430–434. https://doi.org/10.1109/I2CT.2017.8226166.
21. Pal, S.; Hitchens, M.; Varadharajan, V. On the design of security mechanisms for the Internet of Things. 12 2017, pp. 1–6. https://doi.org/10.1109/ICSensT.2017.8304476.
22. Daud, M.; Khan, Q.; Saleem, Y. A study of key technologies for IoT and associated security challenges. 11 2017, pp. 1–6. https://doi.org/10.1109/ISWSN.2017.8250042.
23. Josyula, S.K.; Gupta, D. A new security methodology for internet of things. 5 2017, pp. 613–618. https://doi.org/10.1109/CCAA.2017.8229874.
24. Chatterjee, K.; Gupta, D.; De, A. "A framework for development of secure software". *CSI Transactions on ICT* **2013**, *1*, 143–157. https://doi.org/10.1007/s40012-013-0010-8.
25. Samaila, M.; José, M.; Sequeiros, J.; Freire, M.; Inácio, P. Iot-HarpSecA: A framework for facilitating the design and development of secure IoT devices. 2019. https://doi.org/10.1145/3339252.3340514.
26. Wahab, O.F.A.; Khalaf, A.A.; Hussein, A.I.; Hamed, H.F. Hiding data using efficient combination of RSA cryptography, and compression steganography techniques. *IEEE Access* **2021**, *9*, 31805–31815. https://doi.org/10.1109/ACCESS.2021.3060317.
27. Liao, B.; Ali, Y.; Nazir, S.; He, L.; Khan, H.U. Security Analysis of IoT Devices by Using Mobile Computing: A Systematic Literature Review. *IEEE Access* **2020**, *8*, 120331–120350. https://doi.org/10.1109/ACCESS.2020.3006358.
28. Khan, A.A.; Laghari, A.A.; Shaikh, Z.A.; Dacko-Pikiewicz, Z.; Kot, S. Internet of Things (IoT) Security With Blockchain Technology: A State-of-the-Art Review. *IEEE Access* **2022**, *10*, 122679–122695. https://doi.org/10.1109/ACCESS.2022.3223370.
29. Nandy, T.; Idris, M.Y.I.B.; Noor, R.M.; Kiah, M.L.M.; Lun, L.S.; Juma'At, N.B.A.; Ahmedy, I.; Ghani, N.A.; Bhattacharyya, S. Review on Security of Internet of Things Authentication Mechanism. *IEEE Access* **2019**, *7*, 151054–151089. https://doi.org/10.1109/ACCESS.2019.2947723.
30. Malik, M.; Dutta, M.; Granjal, J. A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things. *IEEE Access* **2019**, *7*, 27443–27464. https://doi.org/10.1109/ACCESS.2019.2900957.
31. Bouzidi, M.; Gupta, N.; Cheikh, F.A.; Shalaginov, A.; Derawi, M. A Novel Architectural Framework on IoT Ecosystem, Security Aspects and Mechanisms: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 101362–101384. https://doi.org/10.1109/ACCESS.2022.3207472.
32. Awan, K.A.; Din, I.U.; Almogren, A.; Guizani, M.; Altameem, A.; Jadoon, S.U. RobustTrust - A Pro-Privacy Robust Distributed Trust Management Mechanism for Internet of Things. *IEEE Access* **2019**, *7*, 62095–62106. https://doi.org/10.1109/ACCESS.2019.2916340.
33. Lawal, M.A.; Shaikh, R.A.; Hassan, S.R. Security Analysis of Network Anomalies Mitigation Schemes in IoT Networks. *IEEE Access* **2020**, *8*, 43355–43374. https://doi.org/10.1109/ACCESS.2020.2976624.

34. Xu, B.; Wang, W.; Hao, Q.; Zhang, Z.; Du, P.; Xia, T.; Li, H.; Wang, X. A Security Design for the Detecting of Buffer Overflow Attacks in IoT Device. *IEEE Access* **2018**, *6*, 72862–72869. https://doi.org/10.1109/ACCESS.2018.2881447.
35. Babar, M.; Khan, F.; Iqbal, W.; Yahya, A.; Arif, F.; Tan, Z.; Chuma, J.M. A secured data management scheme for smart societies in industrial internet of things environment. *IEEE Access* **2018**, *6*, 43088–43099. https://doi.org/10.1109/ACCESS.2018.2861421.
36. José, M. Mapeamento de Requisitos de Segurança à Tecnologia na Internet das Coisas. Master's thesis, Universidade da Beira Interior, Rua Marquês d' Ávila e Bolama, 6201-001 Covilhã, Portugal, 2018.
37. Mohurle, M.; Panchbhai, V.V. Review on realization of AES encryption and decryption with power and area optimization. 7 2016, pp. 1–3. https://doi.org/10.1109/ICPEICES.2016.7853276.
38. Jindal, P.; Singh, B. Performance analysis of modified RC4 encryption algorithm. *International Conference on Recent Advances and Innovations in Engineering, ICRAIE 2014* **2014**. https://doi.org/10.1109/ICRAIE.2014.6909247.
39. Rivest, R.L.; Shamir, A.; Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM* **1978**, *21*, 120–126.
40. Wu, Y.; Wu, X. Implementation of efficient method of RSA key-pair generation algorithm. *Proceedings of the International Symposium on Consumer Electronics, ISCE* **2018**, pp. 72–73. https://doi.org/10.1109/ISCE.2017.8355552.
41. Karakra, A.; Alsadeh, A. A-RSA: Augmented RSA. *Proceedings of 2016 SAI Computing Conference, SAI 2016* **2016**, pp. 1016–1023. https://doi.org/10.1109/SAI.2016.7556103.