Benchmark to determine the best performing encryption system on smart devices

Benchmark para determinar el sistema de cifrado con mejor rendimiento sobre dispositivos inteligentes

Alber Montoya Benitez[1]
Bayron Ospina[2]

[1]Instituto Tecnológico Metropolitano (Colombia). Email: albermontoya@itm.edu.co
ORCID: https://orcid.org/0000-0002-7452-8540

[2]Instituto Tecnológico Metropolitano (Colombia). Email: bayronospina@itm.edu.co
ORCID: https://orcid.org/0000-0002-3993-1430

Received: 09-05-2020 Accepted: 17-09-2020

How to quote: Montoya-Benitez, Alber; Ospina, Bayron (2020). Benchmark to determine the encryption system with the best performance on smart devices. *Informador Técnico*, 84(2), 175 - 191.
https://doi.org/10.23850/22565035.2782

Abstract

Currently, telecommunications and especially mobile communications have taken great importance and relevance in people's daily activities. However, the use of mobile devices has been threatened by the growing wave of attacks and Trojans as malware (e.g., exploits) for information theft, and fire damage. To mitigate these attacks there are data encryption techniques and authentication processes. In this way, a breach of confidentiality and authenticity in communications can be avoided. On the other hand, some of the existing encryption algorithms are insecure and may require high computational costs. This work analyzes the performance of three of the main encryption algorithms defined by the National Institute of Standards and Technology (NIST): Rijndael as Advanced Encryption Standard (AES), Serpent, and Twofish, specifying their main operating characteristics, analyzing performance tests on smart devices, to determine which of these algorithms is the most appropriate to be implemented in each device. Finally, an equation called computational cost is generated, which is a function of RAM, CPU, and battery drain, that analyses for symmetries encryption algorithms that can be performed on similar devices to those treated in this experiment.

Keywords: smart devices; computer security; mobile communications; encryption algorithms; computational cost.

Resumen

Actualmente las telecomunicaciones y especialmente las comunicaciones móviles han tomado gran importancia y relevancia en las actividades cotidianas de las personas. Sin embargo, el uso de dispositivos móviles se ha visto amenazado por la creciente ola de ataques y *malware* tipo troyanos (e.g., *exploits*), para robo de información y daño de archivos. Con el fin de contrarrestar estos ataques, se han creado técnicas de cifrado de datos y procesos de autenticación. De esta manera se puede evitar violación de la confidencialidad y autenticidad en las comunicaciones. Por otra parte, algunos de los algoritmos de cifrado existentes son inseguros y pueden requerir altos costos computacionales. En este trabajo se realizó el análisis del rendimiento de tres de los

principales algoritmos de cifrado definidos por el Instituto Nacional de Estándares y Tecnología (NIST por su sigla en inglés): Rijndael como el Estándar Avanzado de Cifrado (AES por su sigla en inglés), Serpent y Twofish, analizando sus principales características de funcionamiento y realizando pruebas de rendimiento sobre dispositivos inteligentes (smartphones y tablets), con el fin de determinar cuál de estos algoritmos sería el más adecuado para ser implementado en cada equipo. Finalmente, se genera una ecuación llamada costo computacional, que depende de la RAM, CPU y el gasto de batería; con la cual, se pueden realizar análisis para los algoritmos de cifrado simétricos en dispositivos similares a los tratados en este experimento.

Palabras clave: dispositivos inteligentes; seguridad informática; comunicaciones móviles; algoritmos de cifrado; gasto computacional.

1. Introduction

Nowadays mobile devices are required not only in communications but for many daily activities. However, in trends such as BYOD (bring your own device), its use is threatened, due to the presence of a high number of attacks. Authentication and data encryption techniques are protections against such threats, helping to make communications and storage safe and legitimate. In contrast, encryption algorithms demand high computational costs and greatly affect equipment performance. On the other hand, miniaturization has reduced the size of the internal components of these devices through new developments, nanotechnology, and more resistant materials. Reducing the size of the elements causes the batteries to store less energy. Over time, some studies have been carried out on the subject, which are the basis for this work. This paper presents a "benchmark" to identify which would be the best encryption algorithm, in terms of computational cost and security, for two types of mobile equipment. For the experiment, various tests were carried out with different sizes and types of files, obtaining average values that were later analyzed and compared with each other, in order to propose the best options and thus use them in each device. This work involves various CPUs, RAM, and battery drain when running three of the most secure encryption algorithms defined by the Federal Information Processing Standards Publications (FIPS PUBS) 2001.

Unfortunately, attacks on mobile communications are increasing exponentially, during the first half of 2019, more than 440,000 unique users were attacked by financial threats, an increase of 7 % compared to the previous year, also the number of mobile financial attacks in the first half of 2019 was 3,740,378, which is 107 % more than in the first half of the previous year. Currently, only a small percentage of smartphones and tablets have installed any security software (Kupreev; Sidorina; Chebyshev; Kuskov, 2019). Moreover, the use of electronic channels has increased in the financial system, but even a high percentage of users still refuse to use online services because they consider them insecure.

One of the main drawbacks in telecommunications is the implementation of an unencrypted authentication system, or in some cases, the use of weak encryption that can be easy to attack, both of which are inefficient in protecting critical information. For example, the WhatsApp application used the crypt5 or crypt7 system to protect conversations, or WEP authentication on a wireless network, which is very fragile to attacks. Furthermore, the computational cost required by some algorithms affects the running time of the battery. In the literature reviewed we found no method to determine which is the best algorithm for each mobile device based on the CPU and memory features as mentioned above. This document shows a benchmark for encryption algorithms as well as the performance of each on a mobile device.

2. Context

2.1. Threats

There are various types of attacks on the foregoing algorithms. These include the Time-Memory Trade-Off (TMTO), an attack that reduces cryptanalysis time by using data stored in the memory. To counter this attack, Saberi; Shojaie; Salleh; Niknafskermani; Alavi (2012) added an encryption block in the *Advanced Encryption Standard (AES)* message authentication exchange. Another kind of attack that affects these authentication systems are dictionary attacks, widely used against *Wi-Fi Protected Access (WPA)* authentication systems (Shivkumar; Umamaheswari, 2011). Because of the developments of WPA, WPA2, and WPA3, the effectiveness of this attack is minimized due to a dynamic change of the authentication key. The Data Encryption Standard (DES) itself was vulnerable to this attack due to its 56-bit short key, forcing the emergence of 3DES and AES with bit lengths up to 128 and 256 bits.

In common mobile SMS attacks, Trojans are generally used to send *premium* messages from devices, however, their use has been replaced by backdoor software, which was developed to facilitate the ill-intentions of cybercriminals who open additional ports. With a backdoor, it is easy to download malware onto your phone or tablet and makes it easy to steal personal information that can include images, videos, phone numbers, email addresses, and location coordinates (Technology Day IT Now magazine, 2013).

2.2. Protection

Pursuant to the ISO/IEC 27001 Safety Standard, approved and published in October 2005 by the International Organization for Standardization (ISO) and the Electrotechnical Commission (IEC) to provide strong communications security, three characteristics must be met, i.e., availability, confidentiality, and integrity. To meet these parameters, it is necessary to secure authentication as an important element in the security process in three ways: biometrics, smart cards or tokens, and keys. Keys can be transferred using encrypted or unencrypted algorithms. Encryption algorithms are divided into two groups: symmetric and asymmetric. The key algorithms of this study are briefly explained below.

2.2.1. AES - Rijndael

It works at the byte level interpreting bytes as a Galois GF (2^8) body (FIPS PUBS, 2001), and covers entries at the 32-bit level by considering them as polynomials of a grade below 4 with coefficients that are polynomials in GF (2^8) (FIPS PUBS, 2001). In calculations, the grade m irreducible polynomials are used as follows (Equation 1):

$$GF\ (pm)\ =\ \{\lambda 0\ +\ \lambda 1X\ +\ \lambda 2X2\ +\ \cdots \lambda m-1Xm-1;\ \lambda i\ \in\ Zp \qquad (1)$$

Block size and key length can be variable in multiples of 4 bytes. The default keys are 128-, 192-bit, and 256-bit. The structure consists of two stages and, in turn, a series of "rounds" that allow certain changes in the state matrix. They will be briefly explained in Figures 1, 2, 3, and 4.

SubBytes: each byte in the state array is replaced with its entry in an 8-bit search table (Equation 2).

$$S\ =\ bij\ =\ (aij) \qquad (2)$$

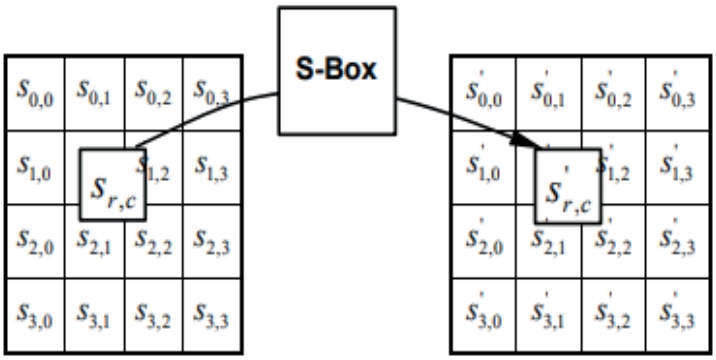

Figure 1. Matrix input replacement
Source: FIPS PUBS (2001).

ShiftRows: the bytes in each row are shifted cyclically to the left.

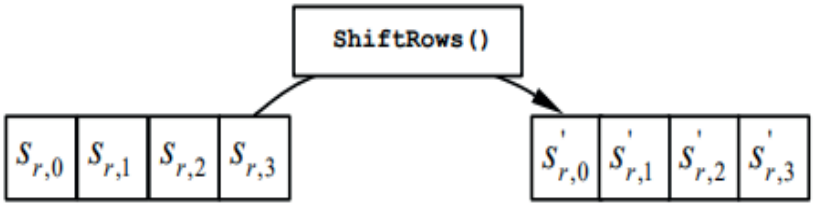

Figure 2. Elements of rotation
Source: FIPS PUBS (2001).

MixColumns: each column is multiplied by a constant polynomial c(x).

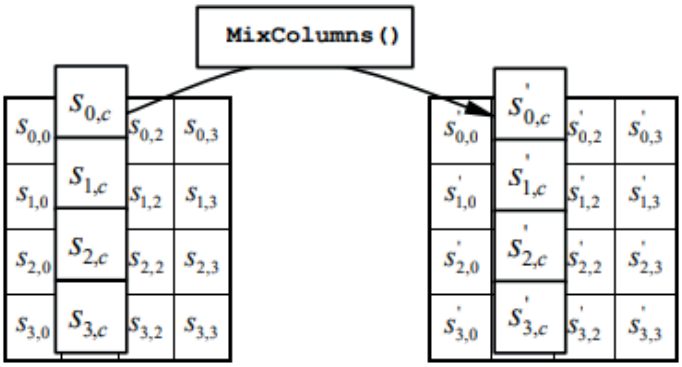

Figure 3. Linear transformation
Source: FIPS PUBS (2001).

AddRoundKey: each byte is combined with one of the subkeys using the XOR $(\oplus)$operation.

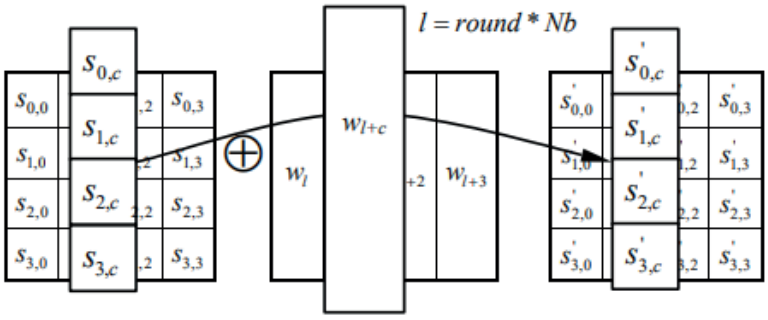

Figure 4. Subkey combination using XOR
Source: FIPS PUBS (2001).

2.2.2. RSA (Rivest, Shamir & Adleman)

Unlike the previous algorithm, this is an asymmetric one, that is applied to both encrypting and digitally signing (RSA Laboratories, 2002). It consists of three steps: key generation, encryption, and decoding.

Key generation

Each user chooses two different prime numbers p and q. (n) is calculated using Equation 3.

$$n = pq \qquad (3)$$

And it is used as a module for keys. $\varphi(n)$ is calculated where φ is Euler's function φ (Equation 4).

$$\varphi(n) = (p-1)(q-1) \qquad (4)$$

A positive integer e minor and relative prime number is chosen for (n). e is shown as the exponent of the public key d, thus d is maintained as the private key exponent (Equation 5).

$$d = e - 1(n) \qquad (5)$$

Encryption

User A communicates his public key (n, e) to user B and A keeps the private key secret. Now B wants to send a message M to A. First, B converts M to an integer m less than n by using a reversible protocol; then calculates the ciphertext c through Equation 6.

$$c \equiv (mod\ n) \qquad (6)$$

Decoding

A can retrieve m from c using its exponent d from the private key (Equation 7).

$$m \equiv (mod\ n) \qquad (7)$$

2.2.3. Serpent

It was designed by Anderson, Biham, and Knudsen (1998). In this algorithm, they used a 128-bit block size that supports key sizes of 128, 192, and 256 bits in length. Encryption involves 32 rounds of substitution-permutation operation in four 32-bit blocks. Each round uses 32 copies of the same 4-bit S-Box. Serpent was designed for operations running in parallel using 32 1-bit offsets (Figure 5).

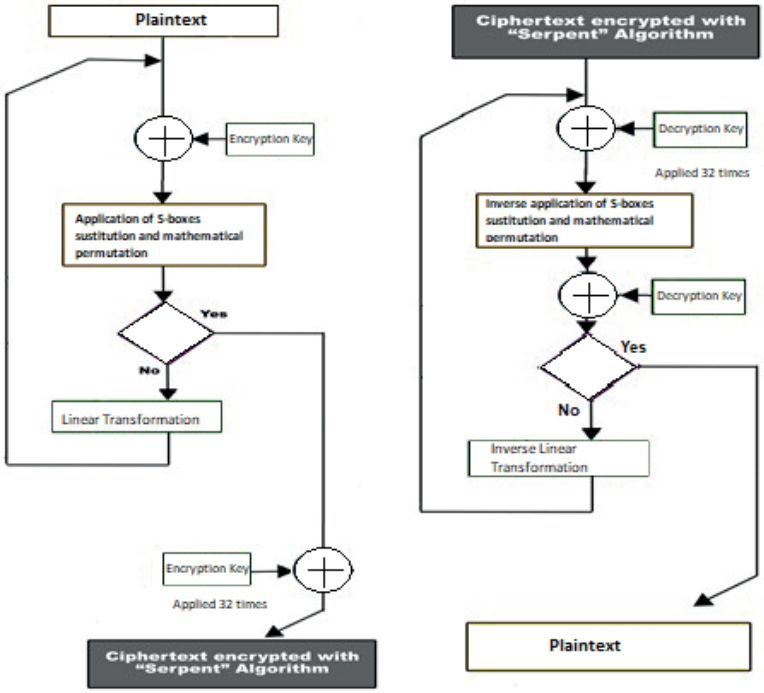

Figure 5. Serpent Cipher
Source: Anderson; Biham; Knudsen (1998).

2.2.4. Twofish

It was designed by Schneier, Kelsey, Withing, Wagner, Hall, and Ferguson (1998). It is a block symmetric encryption method developed by Counterpane Labs and submitted to the NIST contest in search of a replacement as DES. Its block size is 128 bits and the key size can be up to 256 bits. Twofish uses other design elements, such as the SAFER cipher from the Pseudo-Hadamard (PHT) family of transforms. It uses the same Feistel structure of DES (Figure 6). On most software platforms, Twofish is slightly slower than Rijndael for 128-bit keys, but somewhat faster for 256-bit keys (Schneier *et al.*, 1998).

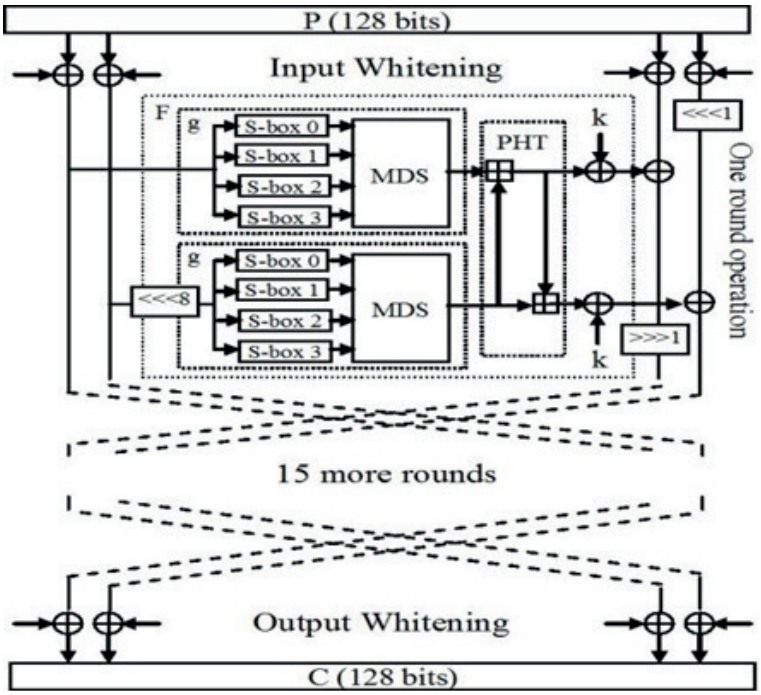

Figure 6. Twofish encryption
Source: taken from Schneier *et al. (*1998).

2.3. Previous research

One of the studies published by the Institute of Electrical and Electronics Engineers (IEEE) proposes a cryptographic security scheme that reduces the use of resources used by mobile devices before uploading data to the cloud. The scheme is the modification of logical and mathematical operations into encrypted blocks (Khan; Kiah; Khan; Madani; Khan, 2013), this proposal may reduce the waste of resources, but could compromise the strength of cryptographic algorithms. In another study, an authentication method is exposed wherein the user's position and time of day are part of the process. In this case, user authentication will be allowed only at certain times of the day. The method is based on the MD5 hash algorithm and DES encryption, but also supports AES and 3DES (Karimi; Kalantari, 2011). This proposal is quite suitable because it increases control and reduces the time of exposure to attacks.

Looking at the current trends in Colombia and the world, multiple developments and applications that can be classified in the Internet of Things (IoT) have taken place. This is the case of Dussán, Vanegas, Chavarro, and Molina (2016), who developed an electronic prototype that monitors physicochemical parameters in order to detect critical situations in fish culture. The prototype developed requires equipment with low machine resources, such as Arduino or the ATMEGA 328P microprocessor, which could require and implement encryption systems for the protection of information.

In a study on VoIP technologies, threat analysis is performed that combines key exchange protocols based on the cryptographic public key Diffie Hellman elliptic curve (ECDH) with authentication based on the user's identity, which also uses the information (identity, IP address, port). The key exchange protocol is proposed to ensure confidentiality and integrity. Bandung and Priyatna (2017) conducted a security analysis between the proposed protocol with the existing

ECDH protocol and compared its key generation performance and key exchange time. The results showed that the combination of ECDH and an authentication mechanism has proven to be secure against

attacks. With the addition of authentication, the total execution time of the build key and exchange key is 11.70 % slower than the original ECDH. However, it can be guaranteed that VoIP communications can still be done interactively. On the other hand, Bian, Lu, and Kuang (2013) propose a modification to the Blowfish algorithm to run on a smartphone. This system improves the efficiency of the algorithm to work on mobile phones, although this is not as safe as the algorithms studied herein that were finalists in the NIST.

Ren, Boukerche, and Nelem (2010) developed a hybrid encryption system with symmetric and asymmetric keys, for authentication in ah-hoc networks, which is still quite secure, although it could be somewhat complex and heavy for certain mobile devices. Similarly, Zhang, Xiao, and Zhang (2010) developed a cryptographic system based on the IDEA algorithm, minimizing the computational cost when running on mobile phones. This could be an interesting development and is an inspiration for current work.

When reviewing research in healthcare, the work by Tovar, Díaz, Quiñones, Pabón, and García (2018) is worth mentioning, wherein a teleoperated physical rehabilitation system is developed for patients living in rural areas. This system handles a set of critical indicators that require low latency to minimize the margin of error in responding to processes that are handled in real-time. Therefore, it is important to deploy software that is low in resource consumption, but that in turn guarantees information confidentiality. Similarly in the review by Castro *et al.* (2017), more developments in the field of health with IoT are presented. On the other hand, in the research of Qi, Pan, and Ding (2011) a *Field Programmable Gate Array (FPGA)* is used, which increases the speed of encryption in short messages on mobile devices, solving the previous problem. This solution can be an example to implement in symmetric algorithms.

On the other hand, Alomari, and Samsudin (2011) implemented an XTS-AES on a GPU. Using parallel processing, they achieved more efficient encryption and with less impact on the devices. Uskov (2012) presents an authentication benchmark and encryption algorithms for MVPN with results that fail to stand up to the algorithms that concern this work. AES was shown to have the best performance on 2 out of 3 platforms tested. Finally, Umaparvathi, and Varughese (2010) performed an analysis of three algorithms: AES, 3DES and Blowfish in MANET, finding that AES is the best. However, 3DES is known to be an obsolete algorithm and Blowfish is less secure than the former.

3. Methodology

3.1. Applied method

The tests consist of measuring the performance of different algorithms when running on certain mobile devices, specifically AES, Twofish, and Serpent; on a smartphone and tablet. Different applications were used, obtaining equivalent results. Based on the inner workings of each application, the following sequence of steps was proposed, where two response variables, CPU and RAM, are considered. The experiment was performed using a technique called randomized block design, which consists on the following steps:

1. Installation and configuration of encryption software (tool 1). In this step, initial statements such as the key size in each application were performed.

2. Installation and configuration of measurement software (tool 2). This tool takes the values of the CPU and RAM variables that display numerical and graphical results in a timeline.

3. Encrypt/decrypt multiple files with different sizes. The tests were performed with different types of files, the formats used were docx, xlsx, jpg, mp3 and avi, which are the most common file types for data storage on the analyzed device.

4. When running the algorithms, only native code was written (without reading or writing to disk). As an alternative test, the algorithms were run in a process that isolates the read and write to the disk, and the applications have the function of explicitly obtaining the consumption of resources (CPU and RAM) from the encryption process.

5. Making measurements of CPU and memory variables to determine battery drain. The values of the CPU and RAM variables were taken when the encryption algorithms were executed. During the measurement process, other external factors were isolated, using a specific starting point as a reference. Finally, the results were tabulated which resulted in an average.

6. Proposing the most appropriate algorithms for each type of mobile device and application analyzed. It consists of proposing a mathematical expression to determine the computational cost of the algorithms analyzed on each device.

4. Results

4.1. Results

Rijndael, Twofish, and Serpent were the algorithms selected for this research, insofar they were the finalists in the AES competition (FIPS PUBS, 2001). Therefore, they are highly used in encryption systems. The tests were conducted on a smartphone and tablet whose hardware features and software are shown in Table 1.

Table 1.
Device Features

SMARTPHONE	TABLET
Samsung Galaxy S10+.	Dell Venue 11 Pro11i-6363blk laptop/tablet
Qualcomm Snap-dragon Octa-core processor 2.4 GHz. 128 GB Flash.	Intel Core i3- 4020Y 1.5GHz processor. Integrated Intel GT2 graphics card. 128GB Solid State Storage Disk.
RAM memory 8 GB.	Ram 4GB DDR3.
Android 9.0 operating system.	Win 10 operating system.
Lithium-ion battery, 4,100 mAh.	37Wh lithium-ion battery.

Source: own elaboration.

Initially, the *Secret Space Encryptor* application was installed on the smartphone, two types of tests were carried out, a *benchmark* for the native code of all 3 algorithms (Figure 7) and the encryption of a 300MB file, which measured CPU performance and RAM usage (Figures 8 and 9). These results will be analyzed graphically below.

Figure 7. Benchmark Smartphone
Source: own elaboration.

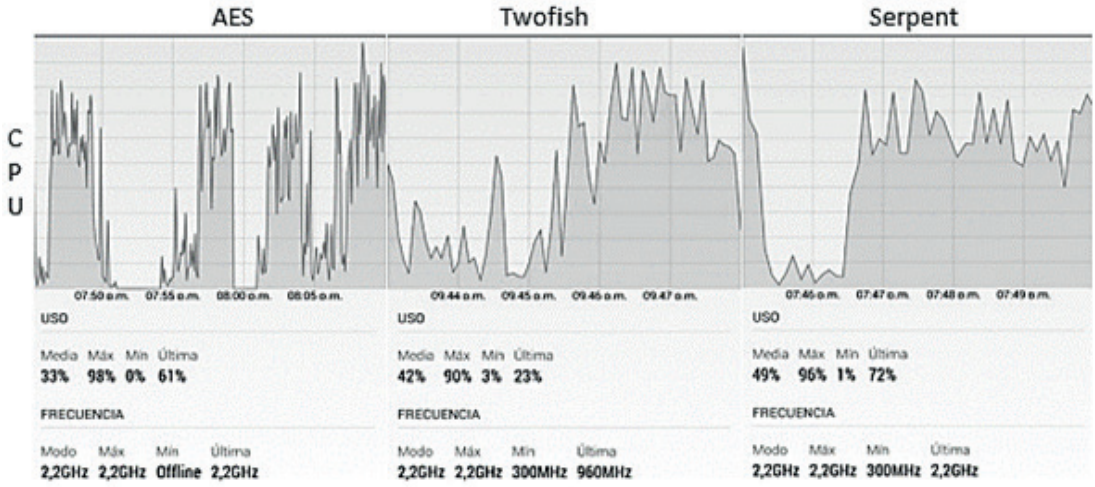

Figure 8. CPU performance in Smartphone
Source: own elaboration.

Figure 9. Use of RAM in Smartphone
Source: own elaboration.

Subsequently, the *TrueCrypt* application was installed on the tablet in which encryption/decryption tests were run, using the same key length, method, and type of algorithm used in the smartphone, as well as possible combinations between peers, for a buffer size of 1GB that isolates the disk write process, insofar it works on a RAM drive. The results obtained by the *software* show that the first place in terms of performance (encryption/decryption process) is for the AES algorithm. Twofish was second, while Serpent was last in performance. The statistical values of the arithmetic means of processing capacity were respectively 518MB/s 80MB/s and 47.1MB/s (Figure 10).

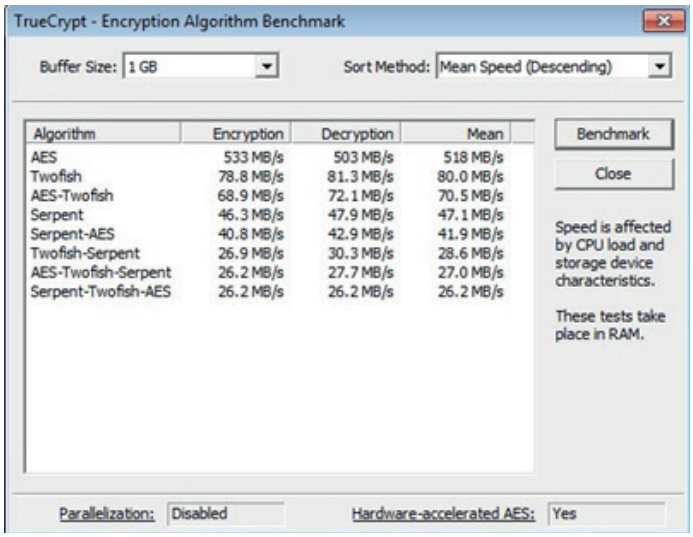

Figure 10. Benchmark on tablet [Mbps]
Source: own elaboration.

In the second type of test, on the tablet, processor performance, memory usage, and write are measured and displayed for each algorithm in the time domain (Figure 11). When running AES, the percentage of processor usage is lower than when running Twofish and Serpent, although memory consumption is lower when running Twofish.

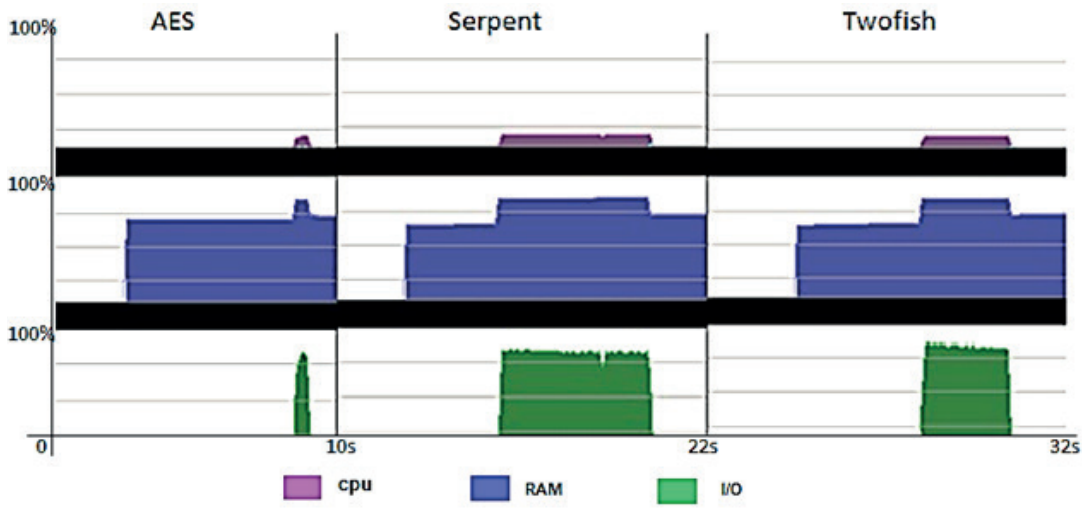

Figure 11. CPU-MEMORY-I/O Performance
Source: own elaboration.

The speed of each algorithm to encrypt the information was obtained using the same tests for each.

4.2. Results analysis

In the first test, the *benchmark* on the smartphone shows the Twofish algorithm as the best performer in terms of machine resource consumption, followed by Rijndael and finally Serpent. The processing speed for each was 942MB/s, 710MB/s and 673MB/s respectively. Meanwhile, regarding the test on the tablet, Rijndael was the best performer with a wide advantage over the other algorithms (Figure 10). This behavior corroborates the theory that the Serpent algorithm is more robust than AES, as it works with more rounds in the encryption/decryption process (32 out of 10) (Anderson; Biham; Knudsen, 1998). This feature improves the cryptographic strength of the algorithm, but makes it slower and heavier to run on mobile devices. As for Twofish, it has a different structure (Feistel), and it is therefore not as strong as AES and Serpent, but it is a more responsive algorithm, therefore, it can execute the cyphering process more quickly than the Serpent algorithm.

In the second test with the Smartphone, a 300 MB file was encrypted. Rijndael performs better in terms of processing, by running with almost 10 % less CPU utilization than Twofish and 16 % less than the Serpent algorithm. In terms of memory consumption, the Twofish obtained less consumption, 33 %, Serpent 35 %, while AES becomes heavier with 39 %. In the case of the tablet test, Twofish did not perform as well, with 54 % CPU usage, only slightly higher than Serpent at 50 % and Rijndael at 46 %. The memory variable test showed that AES consumed less memory than Twofish and Serpent. Table 2 presents a summary of all results and averages.

The analysis of these results enabled us to conclude that they are related to the characteristics of each equipment, that is, that their computing power depends mainly on the operation of the CPU and RAM. However, there are more *hardware* and *software* elements that have an impact to some extent on the internal processes of these devices, for example, the encryption processes. Such characteristics of the equipment are different, since they are manufactured in general for certain specific purposes and, therefore, the proper way to make the best decisions when choosing which equipment to use, is to consider the purpose for which they are acquired. Similarly, the best selection of encryption systems for information protection must be made considering the device used, in order to achieve the best performance results.

Table 2.
Summary of Results

Smartphone	ALG/ VBLE	Benchmark [MB/s]	Speed [MB/s]	CPU [%]	Memory [%]
	Rijndael	710	433	33	39
	Serpent	673	78	49	35
	Twofish	942	96	42	33
Tablet	Rijndael	518	341	46	36
	Serpent	47.1	42.7	50	47
	Twofish	80	70.6	54	39

Source: own elaboration.

In conclusion, as per the theory, all three algorithms are safe, but according to the values obtained in the tests, the AES algorithm has confirmed it has the best performance (encryption rate [MB/s]), low CPU usage, and memory consumption to work on tablets, while Twofish may be a suitable option to implement on smartphones.

4.3. Resource consumption function

Battery drain is directly related to processor and memory usage. Based on these variables, an indicator called computational cost was proposed (whereby we can determine which encryption algorithm has the highest usage of machine resources on each mobile device). Equation 8, describes the cost indicator.

$$I_C = \tau * (0.6\alpha + 0.4\mu) + \frac{\omega\varphi}{\beta}, \quad \varphi > 0, \ \beta > 0 \qquad (8)$$

Where:

I_C: Algorithm Cost Indicator

τ: Average runtime

α: Average of the CPU variable

μ: Average of the RAM variable

ω: Number of rounds of the algorithm

φ: Key length

β: Size of the block to be encrypted

These variables depend on a constant that will give certain relevance to each of them in the computational cost. The relevance was defined at 0.6 for the CPU and 0.4 for the RAM, according to an analysis of variance extracted from the experiment.

Applying this formula to the values obtained from the experiment, the following costs were obtained for each algorithm when running on both devices. The indicator is a value without units (dimensionless), its function is to determine the computational cost (Figures 12 and 13).

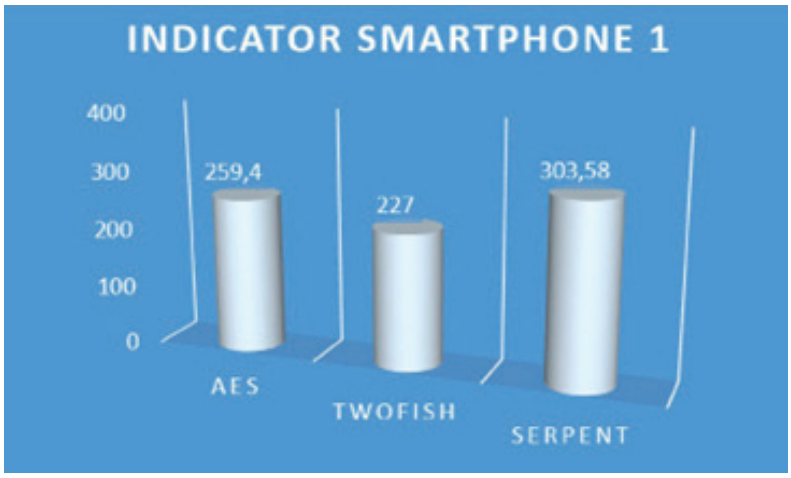

Figure 12. Computational cost of the Smartphone
Source: own elaboration.

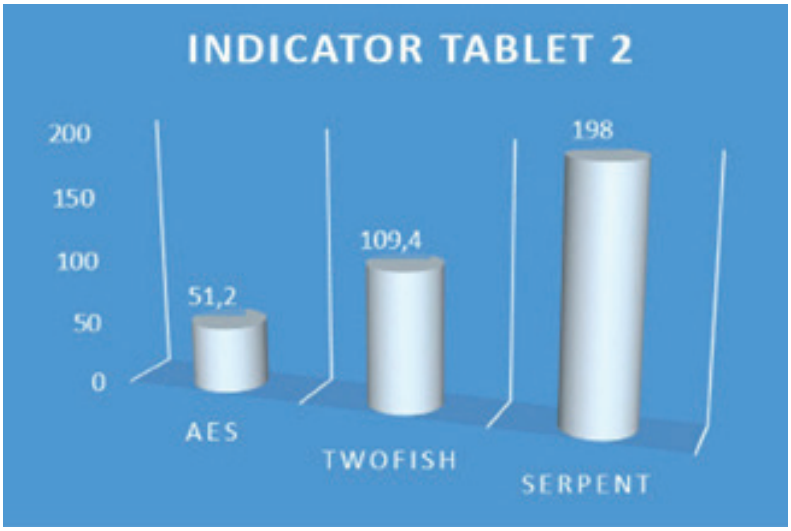

Figure 13. Computational cost of the tablet
Source: own elaboration.

Figure 12 shows that in the smartphone the Twofish algorithm has a lower computational cost than the others, while the results of Figure 13 are applicable to the Tablet, where AES has the lowest computational cost and Serpent has the highest computational cost, compared to the results obtained.

5. Discussion

Considering that there are currently various encryption methods to protect the information that is handled in the different devices, it is important to deploy the best algorithm in terms of performance and energy expenditure in each case. This becomes more relevant in new trends, such as the Internet of Things, where equipment/actuators with low machine resources are managed and, for this reason, their implementations repeatedly fail to use encryption methods as a security measure in their systems. Although the equipment analyzed in this work are *smartphone* and *tablet* devices, the results obtained in the tests and, particularly the computational cost indicator equation, can open a path to also analyze the behavior of the three algorithms on IoT devices.

6. Conclusions

The tests allowed us to determine a "*benchmark*" for the AES-Rijndael, Twofish, and Serpent algorithms. From the results obtained, it can be concluded that AES-Rijndael is the algorithm that requires the least machine resources when the encryption/decryption process is carried out, while the Serpent algorithm uses more resources than the other two. However, the time required for the process execution on the smartphone is longer when using AES-Rijndael.

By including the time variable in the calculations, it was found that AES-Rijndael is the best algorithm to run on the tablet with 50 % of the resources that Twofish would consume, this is a ratio of 2 to 1 in performance. On the other hand, the Twofish algorithm works best on the smartphone with 87 % of what Serpent would consume. This last result is important because Twofish can be presented as a very reliable and safe alternative for this kind of smartphone and could be a strong contender to AES-Rijndael, which is currently the most used.

Moreover, the structure and the number of rounds that the Serpent algorithm executes, make it safe and strong. However, for this reason, it is clear that this algorithm presents slow processing in the mobile equipment evaluated, so it would not be suitable for use in smart devices. Additionally, requiring more machine resources increases battery drain while reducing battery life.

Finally, an equation was developed to estimate the computational cost. This indicator enables us to determine the number of machine resources that each algorithm consumes when running on a mobile device. The equation was elaborated based on time, CPU, RAM, algorithm cycles, key sizes, and blocks to be encrypted. This equation allows the proper choice of encryption algorithms which at the same time is essential for the effective use of cryptography in security.

References

Alomari, Mohammad; Samsudin, Khairulmizam (2011). A framework for GPU-accelerated AES-XTS encryption in mobile devices. *TENCON 2011 - 2011 IEEE Region 10 Conference* (pp.144-148). Bali, Indonesia. 10.1109/TENCON.2011.6129080

Anderson, Ross; Biham, Eli; Knudsen, Lars (1998). Serpent: A New Block Cipher Proposal. In *International Conference on Fast Software Encryption* (pp 222–238). Springer, LNCS.

Bandung, Yoanes; Priyatna, Andri (2017). Development of key exchange protocol to enhance security of voice over internet protocol on mobile phone. *International Journal on Electrical Engineering and Informatics,* 9(1), 173-184. http://dx.doi.org.itm.elogim.com/10.15676/ijeei.2017.9.1.12

Bian, Jiali; Lu, Bei; Kuang, Jian (2012). A new hierarchical file encryption system based on smartphone, Computer Science and Network Technology (ICCSNT). In *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology* (pp. 943-946). Changchun, China. 10.1109/ICCSNT.2012.6526082

Castro, Diego; Coral, William; Cabra, José; Colorado, Julián; Méndez, Diego; Trujillo, Luis (2017). Survey on IoT solutions applied to Healthcare. *DYNA*, 84(203), 192-200. http://dx.doi.org/10.15446/dyna.v84n203.64558

Dussán, Sergio; Vanegas, Oscar; Chavarro, Adrián; Molina, Johan (2016). Diseño e implementación de un prototipo electrónico para monitoreo de parámetros físico-químicos en cultivo de tilapia a través de una aplicación móvil. *Informador Técnico*, 80(1), 49-60. https://doi.org/10.23850/22565035.322

FIPS PUBS (2001). *Announcing the ADVANCED ENCRYPTION STANDARD (AES).* Recuperado de: https://www.cisco.com/c/dam/en/us/products/collateral/security/anyconnect-secure-mobility-client/fips.pdf

ISO/IEC (2005). ISO/IEC 27001: 2005 *Tecnología de la información - Técnicas de seguridad - Especificación para un sistema de gestión de seguridad de la información.* Ginebra, Suiza: ISO/IEC.

Karimi, Rohollah; Kalantari, Mohammad (2011). Enhancing security and confidentiality on mobile devices by location-based data encryption, Networks (ICON), *2011 17th IEEE International Conference on Networks* (pp. 241-245). Singapore, Singapore. 10.1109/ICON.2011.6168482

Khan, Abdul; Kiah, M.; Khan, Samee; Madani, Sajjad; Khan, Atta (2013). A study of incremental cryptography for security schemes in mobile cloud computing environments. In *013 IEEE Symposium on Wireless Technology & Applications (ISWTA)* (pp. 62-67). Kuching, Malaysia. 10.1109/ISWTA.2013.6688818

Kupreev, Oleg; Sidorina, Tatyana; Chebyshev, Victor; Kuskov, Vladimir (2019). *Financial threats in H1 2019.* Recuperado de: https://securelist.com/financial-threats-in-h1-2019/91899/.

RSA Laboratories (2002). *PKCS #1 v2.1: RSA Cryptography Standard.* Recuperado de: https://www.cryptrec.go.jp/en/cryptrec_03_spec_cypherlist_files/PDF/pkcs-1v2-12.pdf

Qi, Na; Pan, Jing; Ding, Qun (2011). The Implementation of FPGA-based RSA Public-key Algorithm and its Application in Mobile-phone SMS Encryption System, In *2011 First International Conference on Instrumentation, Measurement, Computer, Communication and Control* (pp. 700-703). Beijing, China. 10.1109/IMCCC.2011.178

Ren, Yonglin; Boukerche, Azzedine; Nelem, Richard (2010). Performance Evaluation of a Hybrid Cryptosystem with Authentication for Wireless Ad hoc Networks, *IEEE Global Telecommunications Conference GLOBECOM 2010* (pp. 6-10). Miami, FL, USA.

Saberi, Iman; Shojaie, Bahareh; Salleh, Mazleena; Niknafskermani, Mahan; Morteza, Seyyed (2012). Improving confidentiality of AES-CCMP in IEEE 802.11i. In *2012 International Joint Conference on Computer Science and Software Engineering (JCSSE).* (pp. 82-86). Bangkok, Thailand. 10.1109/JCSSE.2012.6261930

Schneier, Bruce; Kelsey, John; Withing, Doug; Wagner, David; Hall, Chris; Ferguson, Niels (1998). *The Twofish Encryption Algorithm: A 128-Bit Block Cipher*. EE.UU.: Wiley.

Shivkumar, S.; Umamaheswari, G. (2011). Performance Comparison of Advanced Encryption Standard (AES) and AES Key Dependent S-Box - Simulation Using MATLAB. In *2011 International Conference on Process Automation, Control and Computing*. (pp.1-6). Coimbatore, India.

Tecnology Day revista IT Now (2013). *Tecnology Day*. Recuperado de: http://revistaitnow.com/2013/05/seguridad/aumentan-ataques-a-dispositivos-moviles/

Tovar, José; Diaz, Juan; Quiñones, Getssy; Pabón, Anabella; García, José (2018). Development of an information system for teleoperated physical rehab care service via Internet. Pilot case: patients with mild knee injury who live in geographically vulnerable zones. *DYNA*, 8(205), 284-293. http://dx.doi.org/10.15446/dyna.v85n204.67961.

Uskov, Alexander (2012). Information Security of IPsec-based Mobile VPN: Authentication and Encryption Algorithms Performance. *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications* (pp.1042-1048). Liverpool, UK. 10.1109/TrustCom.2012.187

Umaparvathi, M.; Varughese, Dharmishtan (2010). Evaluation of symmetric encryption algorithms for MANETs. In 2010 IEEE International Conference on Computational Intelligence and Computing Research (pp.1-3). Coimbatore, India. 10.1109/ICCIC.2010.5705754

Zhang, Wen-Xiang; Xiao, Si-You; Zhang, Yi (2010). Research on Image-Text Encryption Techniques in Mobile Communications, *2010 Second WRI Global Congress on Intelligent Systems* (pp.115-118). Wuhan, China. 10.1109/GCIS.2010.184