

Hardware based cryptography: technological advances for applications in Colombia using embedded systems

Edwar Jacinto Gomez, Caterinne Perilla Gutierrez, Lina Uyasaba Murillo

Technology Facult, Universidad Distrital Francisco José de Caldas, Colombia

Article Info

Article history:

Received Mar 7, 2020

Revised Jul 7, 2020

Accepted Aug 5, 2020

Keywords:

Digital signature

Embedded cryptography

HSM

Microcontroller

Stand-alone

ABSTRACT

To have totally independent systems that offer a sufficient security scheme has become a necessity in Colombia, this because of the proliferation of IoT type systems and similar; In general, it is required to make stand-alone systems totally independent and distributed to offer users a solution to this need, this work offers the analysis and comparison of two security schemes type digital signature and/or hardware security module (HSM) and its variations, made on embedded platforms type microcontroller software, which shows the strategy to provide information protection, In addition, it is analyzed how each implementation was executed, in which devices and metrics of interest, in the first application the cryptography schemes were made using a deep programming that describes the algorithms in C++ language and in the second implementation the use of the dedicated hardware that the embedded platform type microcontroller had is detailed; In both cases, solutions with an acceptable throughput were generated, allowing to obtain comparable solutions and the same style as those made in a PC or similar hardware. On the other hand, an exhaustive review of this type of solutions in the country-region was made, in order to have a reference as to the possible use of this type of applications.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Caterinne Perilla Gutiérrez,

Technology Facult,

Universidad Distrital Francisco Jose de Caldas,

Cl. 68d Bis A Sur #49F-70, Bogotá, D.C, 111921, Colombia.

Email: jcperillag@correo.udistrital.edu.co

1. INTRODUCTION

The way to encrypt data has evolved due to the great technological boom and the various ways to exchange information that we have today, usually the operation of the infrastructure that exists to ensure the authenticity, integrity and confidentiality of information [1] has been facilitated due to the development of applications created by software. Today, with the growth of independent device features such as microcontrollers, development cards, FPGAs [2, 3] or any device with Linux embedded, cryptography tasks can be decentralized and therefore provide security to any required information without the use of PC-type devices, smart phones or similar. This points to the built-in security using low cost and low power consumption devices, which are programmed in different languages and programming levels [4, 5], which allow to decide if the algorithm can be performed in software or hardware, when cryptography requires simple but massive transformations, this decision is convenient to analyze due to the increasing demand of using electronic means where information is exchanged [6].

Over time, hardware security modules, which are cryptographic hardware devices [7], have been developed. They are described as a hardware and software component that is usually added to a PC or server

and provides the least amount of cryptographic work involving encryption and decryption. In Colombia, the regulations established by the superintendence of finance [8] for hardware-based information security include the requirement imposed in numeral 8.2.1.29, which requires the implementation of hardware cryptographic devices with strong encryption algorithms, provided that there is some exchange of data between clients and service providers by any computer medium or the Internet. Despite the fact that the use of electronic commerce in the country has not reached a high level of implementation, with Law 527, Article 2(c), the definition of digital signatures is one of the most widely used applications at the cryptographic level [9], biometric data, passwords or private cryptographic keys are part of the definition of electronic signatures found in Decree 1074 of 2015 [10], as long as these keys are reliable, and through technological neutrality these advances have the same validity as an autograph signature. As a result, it was decided to improve cryptographic applications in order to optimize processes and strengthen data security, since little is known about the Colombian work on implementing hardware keys or security modules based on the Hight algorithm, which at a low cost and with low energy consumption meet the requirements for information protection.

In the country, the existing technological infrastructures of security are really incomplete, in addition the implementation is made thanks to software and hardware mostly foreign and with few standards for the information both for the government and for the citizens; additionally the few policies for the effective protection of databases makes the national security to be at risk. Considering the above, it is evident that the country does not have enough developments of complete cryptographic applications and that implementations are based on complex codes and not very new techniques that make it unnecessary for companies to use them. This is why hardware-based cryptography will be presented as a technological advance in the interaction between the main means of connection and communication in Colombian society, creating awareness of the use of this technology to resolve the level of security, authenticity and reliability support achieved in Colombia, which reaches 74% at the business level [11].

In this document we compare two methods implemented by Colombian students where integrated systems [12, 13] such as MBED and PSOC are used due to their structure, which facilitates programming in C++ code, cost and precision; In addition to the fact that these devices have specific characteristics in terms of response time and development, there are related methodologies and the implementation of encryption, one more compound than the other, emphasizing the most feasible way to encrypt a key [14, 15] and the most practical procedure to do it, either through a Hight algorithm or through a 32-bit system [16] with a hash algorithm, this because it seeks good performance and accuracy in the generation of strong encryption codes. The devices have programmable user interfaces that have allowed progress in the encryption technique by reducing code creation and algorithm collection, as these devices have adapted certain features that simplify this activity. This manuscript shows projects in Colombia that consequently contribute in an academic way with implementations of this type, based on the little projection that this type of technology has in the country and that as a result looks for the way to be able to incorporate technology that supports these needs and at the same time are newer techniques for future developments.

2. PRELIMINARY CONCEPTS

In recent years the need has arisen to create stand-alone applications using devices and mechanisms that allow the execution of algorithms and cryptographic applications with acceptable performance, low cost and power consumption, using the minimum possible memory, the procedure of encryption does not change as shown in Figure 1, which seeks to optimize the process and make it more reliable.

The projects of this type are limited and makes deepen the applications of light cryptography applied to information security problems, in the first instance was made a hardware key application that uses as a basis the Hight algorithm which is one of the lightweigh algorithms with better performance, all this was done in a microcontroller in standard C++ code, which is easily portable to any device of similar characteristics and that supports this programming language and thus is able to make a totally independent stand-alone application that has the ability to perform cryptographic functions and then implement an HSM that meets the needs of embedded applications today.

First of all, it is necessary to work with private key cryptography algorithms, such as the Diffie-Hellman algorithm [17], which allows two parties who know each other to jointly establish a shared secret key on an insecure communication channel. Since the modular operations required in key exchange are a challenge for machines with low memory, an algorithm applying Fermat's theorem was developed to reduce the amount of operations required when performing operations with modular arithmetic. A solution was also implemented to generate pseudo-random number sequences, using a simple linear feedback shift register (LFSR), to point to an array of cells, which are given a non-linear operation by the XOR operation, to calculate the seed with which each of the keys, both private and public, is generated.

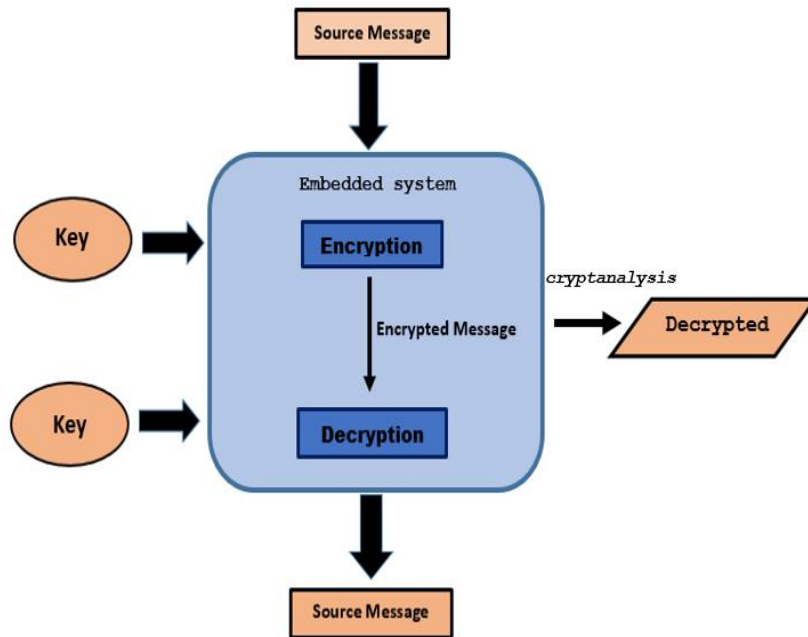


Figure 1. Cryptographic process

The support platforms that were taken into consideration for the project implementation and the implementation of the hight algorithm had to fully comply with the following characteristics: architecture, price, programming language, memory, device speed and ease of acquisition as the Microchip PIC16F/PIC18F/PIC24FJ/PIC32 microcontrollers that are the most used in the country for its easy access [18]; the multiprocessor PROPELLER whose architecture is based on eight processors of 32 bits each and share both output and resource pins [19]; the platform of Texas Instruments that has a family of strong microcontrollers with respect to handling and power consumption (MSP430) and also has under its possession the ARM core license; Cypress Semiconductors company that has the ability to have reconfigurable analog and digital blocks ideal for working with mixed signals and finally Free Scale that has the programming protocol type SDA which allows it to be detected as a mass storage device. This last one was the final choice, the FRDM-KL252Z device that due to the cost/memory/speed ratio is appropriate for minimalist applications, it works with C/C++ language which allows migrating its code to other platforms in an easy way and also the programming of this device is free and open source [18].

The digital signature allows to verify the veracity and authenticity of the origin of certain information [16], uses a combination of public key algorithms, along with asymmetric algorithms, to ensure the veracity of the keys, currently is one of the most used information encryption process, along with the methods mentioned above, uses the benefits of cryptographic functions Hash, as they are algorithms that transform any arbitrary block of data into a new series of characters with a fixed length, with unique characteristics that identify the information to which it was made the digest. It is used to ensure data integrity and authenticity, while storing and authenticating the SHA-256 algorithm [20], which is combined with the RSA asymmetric encryption algorithm to sign and validate the data.

3. IMPLEMENTATION METHODS

It describes step by step how each implementation was done on the devices:

3.1. Method A: Hardware security module (HSM) on microcontroller based on the hight algorithm

For the design of the security module, initially a study of the blocks that compose it was carried out, since the hight algorithm is a block cipher algorithm [21], for this reason, it is necessary that both parties know the main key to achieve the encryption and decryption process. Given that an HSM key has the objective of guaranteeing the security of information, it will have to ensure the knowledge of private keys, using applied mathematical properties that make it difficult to break the security scheme or make the tasks necessary to violate the information computationally impossible, the use of the Diffie-Hellman algorithm, meets the requirements to solve the problem of private key protection in the parties involved.

For the implementation phase the project was developed as follows Figure 2: choice of hardware device, implementation of high encryption algorithm, implementation of high decryption algorithm, creation of high algorithm test interface, standard vector testing with high algorithm, measurement of metrics on the two selected high algorithm MBED devices, implementation of algorithms for secure key exchange for encryption/decryption, creation of test interface of the exchange algorithm, testing of key exchange algorithm, integration of exchange algorithm and high algorithm, creation of HSM end interface, and measurement of end device metrics [18].

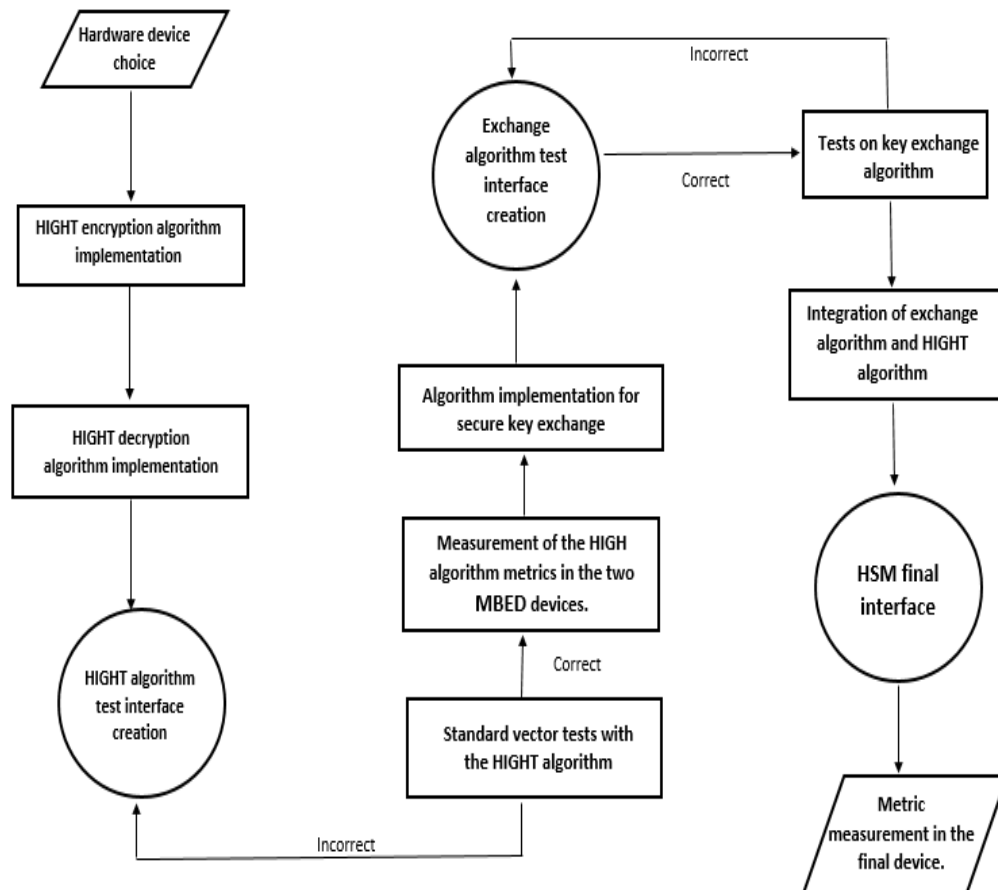


Figure 2. Flowchart for implementation

For the implementation of the algorithm in C++ code, this was done in a standard way, so its migration to other devices did not require executing big changes in the programming, two MBED devices were used: FRDM-KL25Z and NUCLEO-F446RE taking into account specifications, mathematical operations and the characteristics of the hardware device used, the encryption and decryption process were the main blocks. Figure 3 shows the cipher scheme of the High algorithm. Starting with the plain text (64 bits) and the key (128 bits), the Key Schedule function is in charge of generating the Whitening Key used in the initial and final transformation [19]. Then the initial transformation is performed, using the first 4 Whitening Keys generated previously and performing XOR operations and modular addition with the plain text. The last step of the encryption process is the final transformation, this function uses the latest Whitening Key, and the 8-byte text generated and performs XOR and modular addition operations easily implemented in C language, the result of this process is a 64-bit (8 bytes) text called cipher text. While for the decryption block, the following was done:

- The Key Schedule process is performed exactly as in the encryption process.
- The final transformation is applied first and finally the initial transformation, after the 32 rounds the initial one is applied.
- In the opposite direction, the XOR operation is kept the same, although the modular addition is changed by modular subtraction, in C language it is only changing the + sign by -.

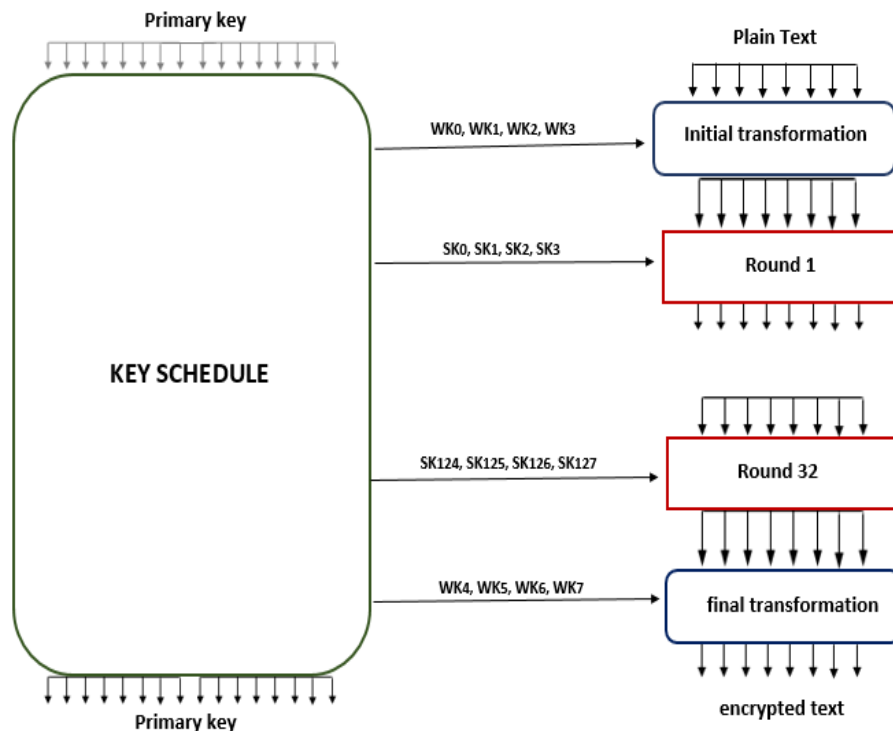


Figure 3. Block diagrams of the cipher

For the graphic development, an interface between the hardware device and the PC through which the user enters the plain text and the secret key, the program created in *Matlab* sends the data to the MBED device to carry out the encryption process and this returns the encrypted text to the interface. The interface is shown in Figure 4. The tests carried out consisted in taking 4 vectors, each one with a 64-bit plain text and a 128-bit key, and the result of encrypting the proposed vectors with the encryption implemented in this project through the interface created was evident. Also at the time of testing the Diffie-Hellman algorithm it was decided to take the number 2 as the generator number because the calculation of powers with this base is facilitated in C++ programming, limiting it to a bit shift to the left and a standard prime number [22].



Figure 4. Graphical interface

3.2. Method B: A digital signature in an embedded 32-bit system

The security scheme of the digital signature is shown in Figure 5, where to generate it a plain text is received, then the Hash algorithm is used to make the summary of the information, and then RSA is used to complete the process of signing the information, when this process is completed the signature is decrypted in the same way using the RSA public key [23], a comparison of the stored message and the summary of the decrypted message is made, if the hash values coincide the validation can be concluded, otherwise the message cannot be authenticated.

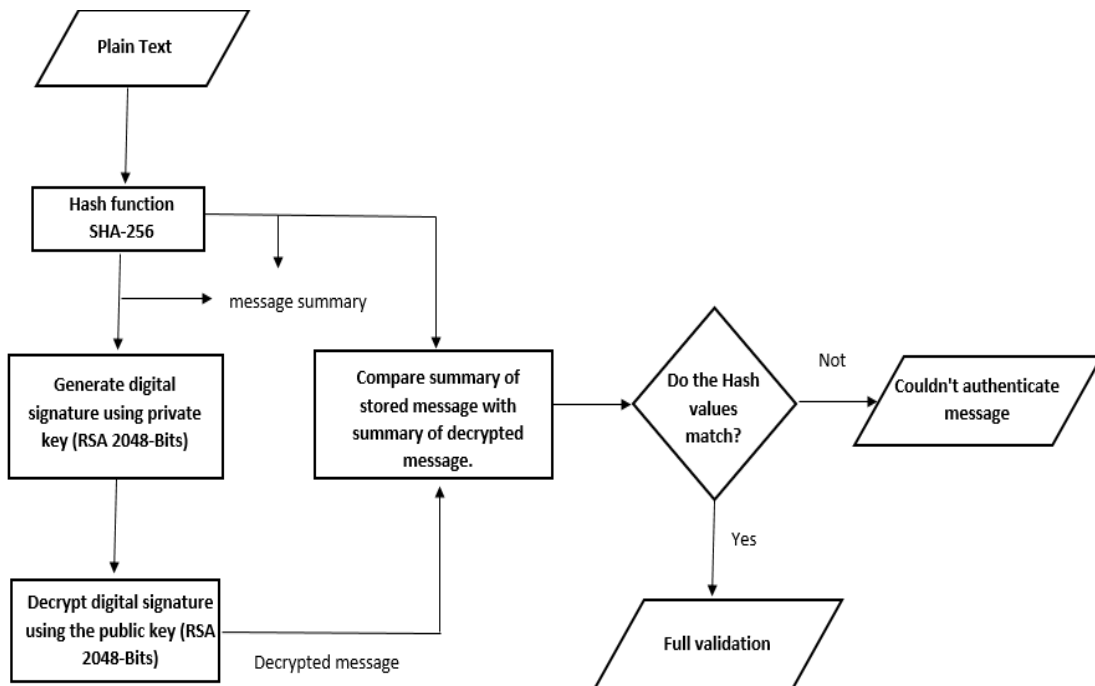
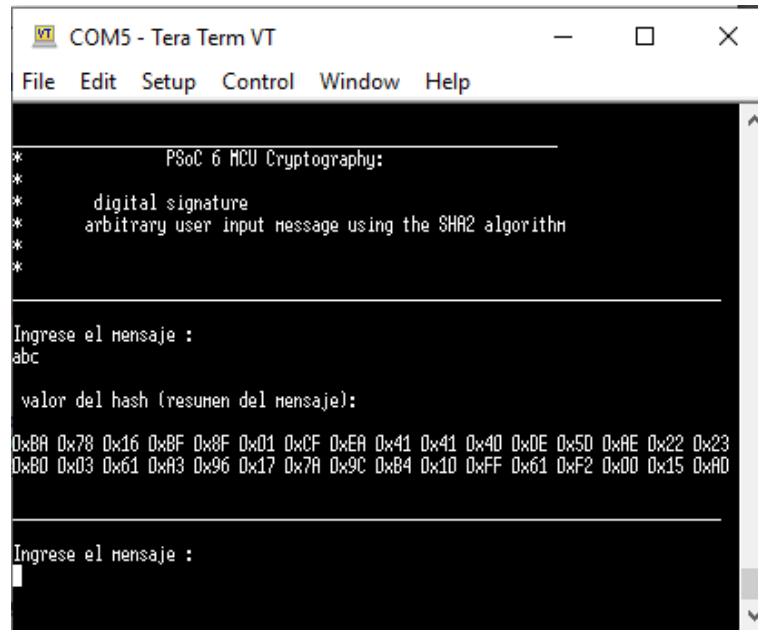


Figure 5. Block diagram for the digital signature

To calculate the SHA-256 compression function, the calculation is divided in two:

- a. Pre-processing of original messages: Involves message stuffing and message scaling for round calculation. Stuffing means adding bits according to some rules until the total length is a 512-bit integer. Then every 512-bit will be expanded to 64×32 bits for SHA-256 round computing.
- b. Then, a digest is done using the dedicated hardware within the embedded platform, in this case the SHA-2 which is a hardware module within the architecture of the PSOC-6 microcontroller, all this is done in a client-server architecture, besides the hardware dedicated to cryptographic tasks this device has two cores a Cortex M0 and a M4, with the possibility of generating independent processing threads. The Cypress IDE allows the use of specialized cryptography libraries, for this application the "cy_crypto.h" and "cy_crypto_server.h" were used, which allow to instantiate the Hash using the dedicated hardware.
- c. Then the signature process is completed, generating the keys with the RSA algorithm under the PKCS standard since it can basically be divided into 4 parts: key generation, encryption, decryption and digital signature. The private key is used to generate the signature and the public key for the respective validation.

Finally, a verification scheme of the application where the message is entered was made, the main function of the program is created that will allow to define if the message is read or not, in case the message is read, the cryptographic block is initialized, the hash is calculated, by means of the `Cy_Crypto_Sha_Run()` function, and it has to be filled in as follows: message read declared as 32-bit integer, message size that by default cannot exceed 100 characters but can be modified, and hash function that will transform the arbitrary block of data into a new fixed-length string [16]. Figure 6 shows one of the tests performed with a Telnet tool, where the hash performed by the microcontroller is verified. Understanding the previous process, the signature is created in PSOC6, where two additional software's are used *Open SSL* and *Python 2.7 or 3*; these to execute a script that generates in the keys generated directory the keys [24].



```

COM5 - Tera Term VT
File Edit Setup Control Window Help
*-----*
*          PSoC 6 MCU Cryptography:
*-----*
*          digital signature
*          arbitrary user input message using the SHA2 algorithm
*-----*
*
Ingrese el mensaje :
abc

valor del hash (resumen del mensaje):
0xBA 0x78 0x16 0xBF 0x8F 0x01 0xCF 0xEA 0x41 0x41 0x40 0xDE 0x50 0xAE 0x22 0x23
0xB0 0x03 0x61 0xA3 0x96 0x17 0x7A 0x9C 0x84 0x10 0xFF 0x61 0xF2 0x00 0x15 0xAD

Ingrese el mensaje :

```

Figure 6. Digest tests performed by the microcontroller shown on a telnet

4. RESULTS AND DISCUSSIONS

4.1. Method A: Hardware safety module (HSM) on microcontroller based on the hight algorithm

The tests carried out consisted in taking 4 vectors, each with a 64-bit plain text and a 128-bit key. The result of encrypting the proposed vectors with the encryption implemented in this project through the interface created was evident as it generates greater understanding to the end user. To optimize the code, at the time of testing the Diffie-Hellman algorithm, it was decided to take the number 2 as the generator number because the calculation of powers with this base is facilitated in the programming of C++, limiting it a little to the left shift and a standard prime number.

When comparing the results, it was discovered that, since the Hight algorithm is optimized for 8-bit integration, certain modifications affecting a higher use of memory resources are required to implement them in 32-bit integrated platforms. However, there are 274x improvements in encryption time and 234x improvements in decryption with the deployment on a card and development platform with extensive and more developed features such as MBED Nucleo-F446RE [18].

The implementation of the HSM module required an extensive study on block ciphers, bit-level operations, key features, discrete logarithms, modular exponentiation, among others. In addition to this, all operations and functions were designed and oriented for the execution of a computer in both hardware and software, thus ensuring the viability of the computer and the efficient use of limited resources such as memory and response time.

4.2. Method B: A digital signature in one system 32-bit embedded

For the implementation of this system it was mainly considered as protecting the keys. The public key is secured either by a method to validate the source of the key, or by using an internal validation with a non-modifiable key that is stored in memory [25]. Finally, after the execution of the code, the response time of the embedded system is analyzed. In the case of PSOC, an internal controller is used that can measure the functions created in the whole structure of the code, with the measurement of these functions the times of the functional blocks are calculated. In this implementation problems were found for the generation and exchange of the files that contained the large prime numbers, generated by the PKCS libraries made by the chip manufacturer, it was decided to export the generated file and import it again through the h-header (header file) that handles the PSOC development card in order to reduce memory and execution time for the code.

In the course of the implementation of the two algorithms, it was difficult to create the hardware key since it was necessary to have a great previous knowledge about codes and cryptographic applications to be able to generate the mathematical operations until obtaining the RSA algorithm. To facilitate the writing of this algorithm, it was decided to use the implementation bases in binary multiplication with vectors that

would reduce the great generation of code, creating a standard function that would perform multiplications and algorithms with 1024-bit numbers.

It is important to note that during the development process were not visible implementations or research known about the use of algorithms type high in hardware security modules (HSM), which makes this a unique research because each of the projects handled information and processes generated by functions in standard C++ programming language, chosen to be adaptable and easy to optimize the hardware, these features make the projects mentioned can be reused in future work, as well as the knowledge gained will make the time spent on these drastically decrease and at the same time encourage research on the subject.

5. CONCLUSION

This paper shows cryptography from another point of view, using dedicated hardware devices, in security schemes adapted to stand-alone systems, which allows protection to be given to a number of possible applications that can be used at many levels such as IoT, HSM applications, sensor networks or any other application that does not require a PC to process the information generated by the processes. In the analysis of the two works exposed, it is observed that in the *method a*: a software embedded system type microcontroller was used; In addition to a High cryptographic algorithm, which, being a lightweight algorithm using simple operations, higher performance, low cost and power consumption, a migration from an 8-bit platform to two 32-bit platforms was performed, a Cortex M0 in which a throughput of 38.1 Kbytes per second (Kbps/s) was achieved and in a Cortex M4 a throughput of 222 kbytes per second (Kbps/s) was achieved by performing all the High algorithm and only using 63% of the RAM of the M0 device and 6% of the program memory.

In *method b*: the PKCS-1 standard is met, which has as standard the implementation of a complete digital signature, this type of implementations are common in applications with PC's, but to make an application of this style in a 32 bits hardware of only a few tens of dollars, makes the system practical and totally stand-alone, since it does not require any additional device; uses Hash functions and the RSA algorithm (which is computationally complex), has a response time of 3.42 seconds, with only a 25Mhz system clock, but the most important thing is that in the consumption parameters a very efficient amount was observed, using 1.3% RAM and 0.2% ROM, since for each of the cryptographic functions the microcontroller has a dedicated hardware with specific memory and resources.

The realization of two totally stand-alone cryptography algorithms, without the need for additional hardware, makes hardware based on cryptography can be used in a number of applications with similar performance to solutions of the same type using PC's; which allows this type of project to be increasingly common in the country, Colombia.

ACKNOWLEDGEMENTS

This work was supported by the Universidad Distrital Francisco José de Caldas, specifically by the research group Seguridad informática Embebida (SIE), a group that in the city, region of Bogotá that is working on the use of dedicated hardware and cryptographic accelerators applicable to local problems, the researchers: Stephanie Sierra Buitrago, Jeremy Aguilar Quiñones and Juan Camilo Ramirez González, are recognized for the realization of such work, which allowed the comparison of methods and strategies applicable to the problems of the country.

REFERENCES

- [1] R.D.V. Velasco, "Criptografía, una necesidad moderna," *Revista Digital Universitaria*, vol. 7, no. 7, pp. 1-9, 2006.
- [2] R. S. Fernandez, "Síntesis de un procesador en VHDL para su posterior volcado en una FPGA," *Politecnica*, pp. 1-199, 2017.
- [3] A. Shiraz, et al., "Single Chip Embedded System Solution: Efficient Resource Utilization by Interfacing LCD through Softcore Processor in Xilinx FPGA," *International Journal of Information Engineering and Electronic Business*, vol. 7, no. 6, pp. 23-27, 2015.
- [4] A. Poschmann, "Lightweight Cryptography: Cryptographic Engineering for a Pervasive World," Dissertation, Ruhr-University Bochum, 2009.
- [5] P. Peris-Lopez, et al., "RFID systems: A survey on security threats and proposed solutions," *11th International Conference on Personal Wireless Communications*, pp. 159-170, 2006.
- [6] V. Cochachin and F. Martin, "Diseño Criptográfico Sobre Hardware Reconfigurable Cifrado Por Bloques Empleando El Estándar De Encriptación Avanzado, Aplicado Al Ámbito Educativo Tecnológico En La Ciudad De Huaraz," Universidad Nacional Santiago Antunez de Mayolo, 2017.
- [7] McAfee, "Needle in a Datastack : The rise of big security data," IT Corporate, p. 9, 2014.

- [8] J. Cifuentes, "Design Of a security management model in wireless communication networks applied to small companies in the private sector of the City of Bogotá," Univ. Nac. Abierta Y A Distancia, no. 3, 2017, [Online]. Available: <https://stadium.unad.edu.co/preview/UNAD.php?url=/bitstream/10596/12862/3/1018402591.pdf>.
- [9] L. Lessig, "La Criptografía Y La Protección a La Información Digital," 2010. [Online]. Available: <https://revistas.uexternado.edu.co/index.php/propin/article/view/2476/3636>.
- [10] Ministerio de Comercio, Industria y Turismo, "Decreto 1074 de 2015," *República de Colombia*, vol. 48, no. 5, pp. 1-405, 2015.
- [11] D. E. Ramírez and H. S. Mora, "Cryptography in cloud computing databases," *Aibi magazine of research, management and engineering*, pp. 49-55, 2014.
- [12] Q. Shi, et al., "FPGA-Based Embedded System Education," *2009 First International Workshop on Education Technology and Computer Science*, Wuhan, Hubei, pp. 123-127, 2009.
- [13] S. N. Soares and F. R. Wagner, "T&D-Bench-Innovative Combined Support for Education and Research in Computer Architecture and Embedded Systems," in *IEEE Transactions on Education*, vol. 54, no. 4, pp. 675-682, 2011.
- [14] A. C. Giménez and M. G. Ibáñez, "Is blockchain technology compatible with the Social and Solidarity Economy? Towards a new paradigm," *CIRIEC-España Rev. Econ. Pública, Soc. y Coop.*, no. 95, pp. 191-215, 2019. [Online]. Available: <https://openaccess.uoc.edu/webapps/o2/bitstream/10609/100826/1/document.pdf>.
- [15] I. G. Roffe, et al., "Implementación del Algoritmo de Cifrado Trivium En Un Sistema Embebido - an Implementation of the Trivium Encryption Algorithm in an Embedded System (in Spanish)," *Pistas Educativas*, vol. 40, no. 130, pp. 573-587, 2018.
- [16] J. Camilo Ramírez González and Ing. Edwar Jacinto Gómez, "Implementation of a Digital Signature in a 32-Bit Embedded System," *Universidad Distrital Francisco Jose De Caldas*, vol. 2003, p. 5, 2007. [Online]. Available: <http://repository.udistrital.edu.co/bitstream/11349/22443/1/Ram%C3%ADrezGonz%C3%A1lezJuanCamilo2019.pdf>.
- [17] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484-1509, 1997.
- [18] S. J. S. Buitrago and J. A. Quinones, "Design and Implementation of a Hardware Safety Module (HSM) on High Algorithm-Based Microcontroller," *Universidad Distrital Francisco Jose De Caldas*, 2017. [Online]. Available: <http://repository.udistrital.edu.co/bitstream/11349/7199/1/SierraBuitragoStephanieJulie2017.pdf>.
- [19] E. J. Gomez, "Performance Analysis of the PRESENT Cryptographic Algorithm using Embedded Hardware and Software Platforms," *Universidad Distrital Francisco Jose De Caldas*, p. 55, 2015. [Online]. Available: <http://repository.udistrital.edu.co/bitstream/11349/2767/1/JacintoG%c3%b3mezEdwar2016.pdf>.
- [20] G. F. Marras, "Hardware architecture and basics of parallel computing," CINECA.
- [21] M. Feldhofer, et al., "Strong authentication for RFID systems using the AES algorithm," *6th International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 3156, pp. 357-370, 2004.
- [22] H. Ellis, "Algorithm Specification," pp. 8-21. [Online]. Available: https://www.csie.ntu.edu.tw/~hsinmu/courses/_media/dsa_13spring/horowitz_28_41.pdf
- [23] RSA Laboratories, "PKCS #1 v2.1: RSA Cryptography Standard," pp 6-8, 2002. [Online]. Available: https://www.cryptrec.go.jp/en/cryptrec_03_spec_cypherlist_files/PDF/pkcs-1v2-12.pdf.
- [24] M. Hastings, "PSOC[®] 3 and PSoC 5LP – Pin Selection for Analog Designs," Cypress, pp. 1-13, 2014.
- [25] A. Karakra and A. Alsadeh, "A-RSA: Augmented RSA," *2016 SAI Computing Conference (SAI 2016)*, pp. 1016-1023, 2016.

BIOGRAPHIES OF AUTHORS



Edwar Jacinto Gomez Received MSc Electronic Control and Instrumentation Engineer in Information and Communications Sciences (Meritorious Thesis). He have more than 9 years of teaching experience in 7 different universities and He currently work as an assistant category teacher at the District University Faculty of Technology, where he formed the RAMA of the District University Faculty of Technology, where he currently the teaching professor, in his teaching he have taught subjects that have to do with the development of software and hardware, especially in tasks of acquisition and signal processing. In the last 5 years, he have published 11 articles in indexed journals, of which 7 are in journals categorized as A2 and A1 by colciencias, on the other hand, he have participated as a speaker in both national and international conferences on at least 20 occasions, as a speaker at least 7 different events of IEEE. He have had a career as a researcher since he was a student in 2002 and up to date, conducting research projects in at least 4 groups of institutionalized research. Of which there have been as results of research, some articles, a research book and several papers of national and international character. He have been a jury of pre-degree thesis on at least 20 different occasions and hesido director of works at this level of at least 25 degree works, of which 3 have been meritorious



Caterinne Perilla Gutierrez Born in the city of Bogota, Colombia. Student of last semester of Engineering in telecommunications, electronic technology of the technological faculty in the district university Francisco Jose de Caldas. Currently She participate in review papers on cryptography and security issues, She works in support and administration of perimeter security services at Claro Colombia.



Lina Uyasaba Murillo Born in the city of Bogota, Colombia. Student of last semester of Engineering in telecommunications, electronic technology of the technological faculty in the district university Francisco Jose de Caldas, with meritorious thesis which was one of the most important projects for the university developed with the Colombian air force. the project called Sonda wise Caldas which amounted to 30 meters high to obtain data from the atmosphere. She currently work as a platform support engineer in the Teleperformance company in Colombia.