PAPER

Lightweight Security Hardware Architecture Using DWT and AES Algorithms

Ignacio ALGREDO-BADILLO[†], *Member*, Francisco R. CASTILLO-SORIA^{††},
Kelsey A. RAMÍREZ-GUTIÉRREZ^{†a)}, Luis MORALES-ROSALES^{†††}, Alejandro MEDINA-SANTIAGO[†],
and Claudia FEREGRINO-URIBE[†], *Nonmembers*

SUMMARY The great increase of the digital communications, where the technological society depends on applications, devices and networks, the security problems motivate different researches for providing algorithms and systems resistant to attacks, and these last need of services of confidentiality, authentication, integrity, etc. This paper proposes the hardware implementation of an steganographic/cryptographic algorithm, which is based on the DWT (Discrete Wavelet Transform) and the AES (Advanced Encryption Standard) cipher algorithm in CBC mode. The proposed scheme takes advantage of a double-security ciphertext, which makes difficult to identify and decipher it. The hardware architecture reports a high efficiency (182.2 bps/slice and 85.2 bps/LUT) and low hardware resources consumption (867 slices and 1853 LUTs), where several parallel implementations can improve the throughput (0.162 Mbps) for processing large amounts of data.

key words: hardware, steganography, cryptography, DWT, FPGA

1. Introduction

The information age is currently changing on a daily basis, moving from a primarily industrial culture to a communications one. Information is used by people through the internet, cell phones, and social networks, generating valuable components. The problem occurs when this information, either personal or confidential, is intercepted and read causing different types of risks as damaging its own integrity or the success of a project.

Sharing information in industry over the internet has become an essential part of inner processes, from sharing informative emails to innovative details regarding new processes, products, etc. On the automotive industry, sharing information for new developments happens among many divisions in different parts of the world, as production and sale/marketing departments. Threats against automobiles and infrastructure such as power plants are focused on. Security modules including cryptosystems begin to be introduced in embedded systems used in these fields [1]. Spec-

ifications for new models are classified data until the car is exhibited in a special auto show. An important part of the development of a new car is the styling design, which includes data on overall height, length/width, wheelbase, ground, track coefficient of drag, head room, shoulder room, leg room, seating capacity and so on; the majority of this data has to remain confidential, because it can be reached by unauthorized users, requiring security services such as authentication, integrity, non-repudiation, privacy, etc.

There are several methods to secure the information transmission through non-secure channels, among these, the most important are cryptography and steganography; the first one scrambles a message so that it cannot be understood, while the second intends to hide the existence of the message [2]. In general, a steganographic system consists of three elements: 1) cover object, which hides the secret object, 2) the secret object and 3) the stego object, which is the cover object with the secret object embedded inside it [3]. Steganography sends a secret object under the camouflage of a carrier content. The carrier content appears to have totally different but normal (“innocent”) meanings.

Both cryptography and steganography can be implemented in software. However, these algorithms could become very complicated and would lead to a significant use of computational resources, both spatial and temporal, that is why a hardware implementation is a better alternative. Integrated circuits (IC) nowadays play crucial roles for critical applications in the automotive, aviation and medical fields [4]. This research implements steganographic operations in hardware, using the DWT [5] combined with an AES cryptographic algorithm [6]. This enables the advantages of hiding a ciphertext on cover objects, where if this is attacked, the plaintext is ciphered. On the other hand, when the existence of ciphertext is known, this can motivate attacks. The proposal provides a total dispersion of the ciphertext, which is the hidden information on the stego object. In this way, the secret object is not easy to obtain when the carrier is attacked.

This paper is organized as follows: in Sect. 2, the lightweight security solutions are presented and described, whereas in Sect. 3, DWT and AES algorithm are detailed. In Sect. 4, the proposed system is presented, and in Sect. 5, hardware implementation of the proposal and its main modules are described. The results obtained from the simulation using the proposed hardware architecture and the compar-

Manuscript received May 17, 2018.

Manuscript publicized August 2, 2018.

[†]The authors are with the Computer Science Department, CONACYT-National Institute of Astrophysics, Optics and Electronics, 72840 Puebla, México.

^{††}The author is with the Telecommunication Engineering Department, Autonomous University of San Luis Potosí, 78290 San Luis Potosí, México.

^{†††}The author is with the Civil Engineering Faculty, CONACYT-Michoacan University of Saint Nicholas of Hidalgo, 58240 Michoacán, México.

a) E-mail: kramirez@inaoep.mx

DOI: 10.1587/transinf.2018EDP7174

isons are given in Sect. 6. Finally, Sect. 7 draws the conclusions.

2. Lightweight Security Solutions

In the modern world, three main emerging concepts (security, mobility and connectivity) generate new applications or update the existing ones. Therefore, in the area of the computational design, these concepts require wireless devices that allow the proliferation of communication systems and mobile computing, bringing revolutionary changes to the information society. There are many examples of technological advances: i) devices such as smartphones, laptops, tablets and others, and ii) systems such as connected vehicles, autonomous automobiles, home automation, industry 4.0, IoT-based systems, and so on.

The security of information is essential because third parties can cause problems as: i) modification attacks, where a transmitted message is modified and sent to the destination; ii) fabrication attacks, where a message is inserted into communication channels and networks; iii) interception attacks, where a message is intercepted by an unauthorized individual, and iv) interruption attacks, where a message is hampered by some communication process between one or more systems. Different security services must be provided to communication devices and applications to protect them against unauthorized users, preventing systems from becoming unusable, avoiding loss of confidentiality, receiving inaccurate messages, etc. Cryptography is one of the main mechanisms to reinforce the level of security and is based on complex computational algorithms, which use high computational costs (time and area). This motivates the work in new lines of research to design and develop software and hardware architectures that combine and provide several features, such as: low power consumption, small size (low consumption of hardware resources), high performance and high efficiency. A system must reach several requirements that implies a complex design process, since multiple trade-offs must be analyzed.

Lightweight security solutions can be applied to programs, devices or architectures, which are simpler, faster or smaller. These solutions are being implemented in new applications such as integrated systems, Internet of Things, mobile devices, automotive systems, etc., where low power consumption and high performance are required. Although there are subsystems such as standby power (power consumption when electronic devices are off at the electrical interface) or inactive tasks (methods to reduce internal processes and consequently, power consumption), new architectures must still be developed to save energy.

It is important to note that the total power energy consumption consists of two types: dynamic power and static power [7]; the first one is mainly based on the transitions of the clock and is controlled in the design stage or in the execution time, but the second is not controlled, since it is consumed independently of the execution of some activity or task. In this case, small hardware architectures can re-

duce power consumption. This is due to the physical design, where the smaller area of silicon requires less energy consumption, saving batteries in mobile or automotive applications. However, certain components (malicious circuits, large embedded systems, large number of sensors, and others) can exhaust or age a car battery under certain conditions or can be backdoor and send information to third parties [8], which are not desired features.

3. DWT and AES Algorithm

In this section, two important algorithms are presented, which through the years have been widely implemented on several applications and are used in the proposed system: the DWT and AES.

3.1 Discrete Wavelet Transform (DWT)

The DWT is widely used in many applications as data compression [9], object recognition and watermarking [10]. There are several steganographic algorithms such as [11]: *blindhide*, *hide seek*, *filter first*, *battle steg*, etc; but algorithms based on DWT are important and emerging solutions. This transformation is discrete in time and scale, in other words, DWT coefficients may have real (floating-point) values, but the time and scale values used to index these coefficients are integers. For this application the 2D Discrete Wavelet Transform is utilized, this transformation can be done in two ways; first by applying the low- and high-pass filters along the rows of the data and later applying each of these filters along the columns of previous results, and the second way is to apply four matrix convolutions, one for each low-pass/high-pass, horizontal/vertical combination. The 2D transform is separable which means that is an application of the 1D DWT in the horizontal and vertical directions [12]. An example of how an image is decomposed by the DWT is shown in Fig. 1.

3.2 Advanced Encrypted Standard (AES)

AES is a specification to cipher electronic data, established in 2010 by the U.S. National Institute of Standards and

Fig. 1 Example of 2D wavelet decomposition.

Fig. 2 AES algorithm.

Technology (NIST) [13]. The AES algorithm is a symmetric block cipher, meaning it uses the same key to cipher and decipher, this algorithm use cipher keys of 128, 192, and 256 bits. The plaintext is divided into data blocks, which are processed during a number of rounds using substitutions and permutations. The plaintext is the data (file contents or binary files) that represent only bits or characters of readable material (e.g. integers, text, images, audio, etc.) but not graphical or audible representation nor other objects (images, videos, documents, etc.). The AES algorithm is resistant to attacks, its code is fast and compact on various platforms and its design is simple.

The structure for AES is shown in Fig. 2, more details in [14]. The algorithm begins with an initial round (*AddRoundKey*), then nine rounds of four functions and the tenth round of three functions. The rounds of four functions are composed by: Non-linear byte substitution (*SubByte*), Shift Row Transformation (*ShiftRow*), constant Galois field multiplication (*MixColumn*), and Key Addition (*AddRoundKey*). The tenth round does not incorporate the Mix Column function. After these rounds are performed, the ciphertext is obtained [15]. The ciphertext is the data obtained from plaintext after applying a cryptographic algorithm, called cipher. Deciphering is the inverse operation of ciphering and is the process of turning ciphertext into original plaintext.

For this application the Cipher Block Chaining (CBC) Mode of the AES algorithm is implemented. It produces different ciphertext blocks from the same plaintext blocks, this is achieved by XORing each plaintext with the ciphertext from the previous round; on the first round uses an Initialization Vector.

4. Proposed System

There are several techniques for hardware design, such as

Fig. 3 Block diagram of steganographic/cryptographic algorithm: (a) transmitter side and (b) receiver side.

unrolling, pipelining, iterative structures, hardware reusing, etc, and the mobile or automotive applications present diverse limitations that must be taken into account during the architecture design, and these applications use limited power sources. Therefore it is complex to provide solutions for these constrained environments, because balancing analysis among several aspects of the area, power consumption, throughput, and so on, must be made. In this proposal, an iterative architecture is reported, which reuses one multiplier and one adder for computing the $2(2n^3 - n^2)$ operations, see Sect. 5, whereas AES module reuses one round for computing the 10 rounds defined in this algorithm for 128-bit keys and 128-bit blocks.

The block diagram of the proposed system is shown in Fig. 3. Firstly, the *plaintext*, which can be some file, image or sound, is ciphered by the AES module, generating *ciphertext*. Secondly, the *cover object* is generated, where the ciphertext will be hidden. Then, the *Discrete Wavelet Transform (DWT)* is applied to the cover image; it transforms data from space to frequency domain and its main objective is to know the repetition number of an event. For example, if the cover object is an image, this process guarantees that changes will remain imperceptible to the human eye. As a result of the transformation, a matrix is obtained, where the majority of the coefficients are zeros, considered as spaces where data will be hidden; later this matrix will be processed by the *thresholding* method, where they are compared with a value K and all coefficients under this value will change to zero, generating a storage space. The transformed image

after applying the thresholding conserves low distortion and contains a large number of zeros. The ciphertext is stored by the *Steganographic Function* on the spaces created before. After, *Inverse Discrete Wavelet Transform (IDWT)* is performed, generating the *stego object*, which has the ciphertext hidden on it, see Fig. 3 (a).

The main task is to retrieve the secret object embedded in the stego object. The thresholding method gives a space to hide the secret object in the DWT domain. In principle, the secret object can be retrieved from these cells (spaces with 4 LSBs) corresponding to the bytes with values less than or equal to the threshold K in the DWT domain. After embedding the secret object (ciphertext), the IDWT is applied, then, the obtained object must be normalized and rounded. These last processes can adjust the stego object to valid values according to the format, however, these processes can introduce an irretrievable distortion in the secret object. It is important to remember that the stego object can be any document or file as text, audio, video, etc. The maximum distortion δ can be measured from a bank of objects of the application. Then, the secret object (in the proposal, it is the ciphertext) can be retrieved without distortion from the values less than or equal to $K - \delta$, after DWT is applied to the stego object. The modulation of the secret object is required to map binary sequences of the same object to an M -ary constellation with amplitude values less than the selected threshold ($K - \delta$) and separate a distance $\Delta > \delta$. So, the modulated signal is inserted in the spaces of the transformed cover object using one symbol per byte. Simulation results show the effectiveness of the proposal.

On the receiver side, the DWT is applied to the stego object to retrieve the embedded object (ciphertext), see Fig. 3 (b). After performing this process, $a(k)$ values should be obtained, which has an absolute value less than or equal to the threshold minus the distortion factor δ , $a(k) = |G(i, j)| < (K - \delta)$. After, the ciphertext is retrieved by using the signal from $a(k)$ and the mapping rule in the M -ary constellation. Finally, the ciphertext is deciphered by using the AES algorithm, which produces the deciphered text or plaintext.

5. Hardware Implementation

A variation of the discrete wavelet transform is the Haar wavelet. This transform is widely used to hide information on images. The Haar wavelet is defined as:

$$h_n(X) = \frac{1}{\sqrt{N}} \begin{cases} 2^{\frac{j}{2}} & \frac{k-1}{2^j} \leq x < \frac{k-1/2}{2^j} \\ -2^{\frac{j}{2}} & \frac{k-1/2}{2^j} \leq x < \frac{k}{2^j} \\ 0 & \text{otherwise} \end{cases}$$

The integer j determines a binary dilation by 2^j , while k determines the translation by $k/2^j$, $n = 2^j + k - 1$ for $j = 0, 1, \dots$; $k = 0, 1, \dots, 2^j$ and $N = 2^n$. In the proposed architecture, the associated matrix for h_n is represented by H of size $N \times N$. Since the Haar transform is separable, then $H^{-1} = H^T$ and since H is not symmetric, H^T can not be replaced by H , so if F is the matrix of the image and G is the resulting transform of $N \times N$, then Eq. (1) is obtained for transmitter side.

$$G = HFH^T \quad (1)$$

On the receiver side, the application of H^T on the left side of the Eq. (1) and H on the right side, Eq. (2) is obtained.

$$F = H^T GH \quad (2)$$

To develop Eqs. (1) and (2), that represent the middle part of the steganographic algorithm, the proposed hardware system is shown in Fig. 4.

In Eq. (1), three variables are used, where H represents the Haar Matrix, H^T is its transposed matrix and the variable F is the cover object, which is provided externally and stored in the internal memory. The block diagram of the proposed system is shown in Fig. 4, where *Address generator* module provides the different storage locations in the memories through parallel communication, not directly from the computer.

In this way, each memory location represents an element of the matrix, so the zero location in memory represents the element h_{11} of the matrix H , the location 1 represents the element h_{12} and so on. Equations (1) and (2)

Fig. 4 Steganographic system implemented in hardware using plaintext and cover image.

indicate multiplication of matrices, from the hardware perspective, these operations are about multiplication between data H and F , and the *Address generator* produces different address to store the h_{in} data belonging to row i of H and the component f_{nj} belonging to column j of F to perform Eq. (3). If H has a size of $m \times n$ and F has a size of $n \times p$, then $M = HF$ matrix for the first multiplication has a size of $m \times p$. In this way, this multiplication requires to obtain each coefficient of M as is shown in Eq. (3).

$$\begin{aligned} m_{ij} &= (\text{row } i \text{ of } H) \times (\text{column } j \text{ of } F) \\ &= h_{i1}f_{1j} + h_{i2}f_{2j} + \dots + h_{in}f_{nj} \end{aligned} \quad (3)$$

At this point, each dot product executed by the *Multiplier* between the corresponding coefficient of row i of H and corresponding coefficient of the column j of F is stored in the *Register1*. After each dot product, the *Adder* module operates the data stored in *Register1* and *Register2*, this last is an accumulator for adding the dot products, which are necessary for finding each coefficient m of the matrix M . The process is repeated until the multiplication of the last element of row i of H with the last element of column j of F is executed, storing this value in *Register2*. The buffers found in the output of the *Adder* and *Multiplier* allow the protection of these modules, preventing a virtual short circuit from occurring. The previous process computes the operation HF , which is the product of the Haar matrix and the cover object. The magnitudes for each coefficient of the Haar matrix, the transposed matrix and the matrix of the cover object are floating-point magnitudes based on the IEEE 754 standard and require 32-bit buses. The process ends until the operation between the last row and the last column has been completed, storing all values in the $m \times p$ locations, so HF will be stored in the memory from location 0 to $m \times p - 1$, which is the first half of the internal memory.

The following process is the matrix $M = HF$ multiplied by H^T to obtain $M \times H^T = HFH^T$, which is stored in the second half of the internal memory, performing at the same time the thresholding technique. If the *Comparator* module detects that the coefficient is smaller or equal than a constant K , it allows the passing of the variable *ciphertext*. This last was previously processed in parallel by AES module, more details in [14], where the input is *plaintext*. Finally, the matrix HFH^T is sent by the transmitter. An important contribution is that the same architecture functions on both sides: transmitter and receiver. In the receiver side, the transmitted message is stored in the same order with the objective of retrieving the information in subsequent processes. To retrieve the ciphertext, an inverse process is implemented using Eq. (2), where the first product $H^T G$ is stored in the first half of the internal memory, and the second half has stored the received matrix HFH^T that represents the matrix G .

The received coefficients are multiplied by the coefficients of previously stored matrix H^T . Finally, the product $H^T G$ is stored in the second half of the internal memory, the activation of a buffer will allow the reading for a subsequent procedure, either by reading the information from the com-

puter or by storing it in a memory. These data represent the ciphertext which is deciphered by using AES cryptographic module, which provides the plaintext.

The hardware implementation can be used for both transmitter side and receiver side, which is due to the architecture design. As, the secret object can be text, audio, image, video, or some digital file. In this work, two examples are described and hidden: text (vehicular data from OBD-II) and image (design of a car), which are operated in binary format and are the plaintext.

In the transmitter side, two concurrent processes are executed: AES and DWT algorithms. The first one takes the plaintext, divides it into 128-bit blocks and ciphers each block to produce cipher-blocks (all cipher-blocks concatenated are the ciphertext). The second one applies direct DWT on the cover object F , which is the HF matrix, and each value cell is compared with a threshold value (this last can be obtained from an extra analysis), and in this work, for example, $K = 15_{10} = 0F_{16} = 00001111_2$. If the cell value is less than or equal to K , then its value is $00_{16} = 00000000_2$, making room for data hiding, and four LSB (least significant bits) can be used in this process. For this case, each half of byte from the 128 bits of the cipher-block is dispersed in the four LSBs, allowing to hide bits of the ciphertext. This new value cell is very similar than the previous value because thresholding transforms from a set of values (from $00_{16} = 00000000_2$ to $0F_{16} = 00001111_2$) to another set of values (a set of values from $00_{16} = 00000000_2$ to $0F_{16} = 00001111_2$), where the graphical result will show a stego image similar than the cover image. The new HF matrix obtained from cover object and ciphertext (original FH matrix) has: a) not-modified value cells, where their values are greater than K , and b) modified value cells, where their values are less than or equal to K , because of the 4 LSBs have 4 bytes from cipher-blocks (this decision from some K value enables the receptor for recovering the cipher-blocks). So, the new FH matrix is processed by the inverse DWT (IDWT), which is reached by the matrix multiplication between the new HF matrix and H^T matrix, obtaining the G matrix ($G = HFH^T$), which is transmitted.

In the receiver side, the stego object G is stored. So, two processes are applied to recover the plaintext. Mainly, the first one only computes the direct DWT, using threshold value K for recovering the ciphertext. In this case, ciphertext is retrieved by the 4 LSBs of the value cells less than or equal to K . After, the second one takes ciphertext and decipheres it. Finally, the deciphered object is presented, which is the plaintext.

6. Results

The implementation results are based on Xilinx ISE version 14.7 and the development board named Automotive Spartan 6 XA6SLX100. Table 1 presents the comparison of the proposed system with previous researches. The designed system uses a smaller number of slices-registers and slices-LUTs, allowing a greater efficiency in the use of resources,

Table 1 Related works on different boards.

Authors	Development board	Slices	Slices LUTs	FrEq. (MHz)
A.A. Jadhav <i>et al.</i> [16]	Spartan 3EDK	1850	2985	72.49
C.S. Maya <i>et al.</i> [17]	Spartan 3EDK	1475	2330	294.291
A.F. Mahmood [18]	Virtex-II XC2VP30	2411	-	107.75
K.N. Pansare <i>et al.</i> [19]	Spartan 2 x2s100	1880	2971	20.487
B.J. Mohd <i>et al.</i> [20]	Cyclone II (Altera)	1572	-	72.2
Proposed System	Spartan6 XA6SLX100	867	1853	40.54

Table 2 Results of the hardware implementation.

Frequency/Period	F = 41.49 MHz/24.101 ns
Throughput	162.90 Kbps
No. of Slices	825 (381 DWT + 444 AES)
No. of Slices-LUTs	1890 (1021 DWT + 869 AES)
Efficiency (Registers)	197.48 bps/reg
Efficiency (LUTs)	86.10 bps/LUT

and its operation frequency is lower than the others, which means that the system has a smaller clock frequency except for the proposed by [19], and together with the area, they reduce the power consumption.

The innovation and contribution of this work is the proposed architecture, where the algorithm can multiply square matrices of any magnitude allowing the use of shared internal memory. Another contribution is the storage of the products described in Eqs. (1) and (2); initially four RAMs were used, allowing the storage of HF and HFH^T to find G , as well as H^TG and H^TGH to find F . This is optimized replacing those four memories by a single one and dividing it in two, where the first half stores the H^TG product while the second half stores the H^TG result corresponding to G . A similar situation occurs on the receiver side.

For implementation results, this architecture was evaluated by using a cover image with a size of $256 \times 256 \times 32$ bits = $2'097'152$ bits, and a text to hide (ciphertext) of $16'384 \times 8$ bits = $4'096 \times 32$ bits = $131'072$ bits. In this case, the text is data from vehicle's OBD system (On-Board Diagnostic system) in 8-bit ASCII code, which is arranged on 32-bit strings. This 4096-byte text is ciphered to obtain 4096-byte ciphertext, which was processed by the AES as $1'024$ block of 128 bits ($1'024 \times 128$ bits = $131'072$ bits), and each block requires a 10-cycle latency, reporting $10'240$ cycles for the complete plaintext. At the same time, the matrix multiplications for the stego algorithm are executed. The multiplication for two $n \times n$ square matrices requires n^3 multiplications and $n^2(n-1)$ additions, and there are two matrix multiplications: $H \times F$ and $(HF) \times H^T$, resulting in a count of $2(n^3 + (n^2(n-1))) = 2(2n^3 - n^2)$ operations. If the cover image has a size of $256 \times 256 \times \text{pixel of } 32$ bits, then $n = 256$ and $33'423'360$ operations are required. Finally, initializing tasks, storing and addressing memories/registers and sequencing of multiplication/adder require $538'247'159$ clock cycles for computing stego process HFH^T , where the cipher process AES is executed in parallel. The hardware architecture reports a throughput of 162.9 Kbps operating at 41.49 MHz, see Table 2.

Additionally, a graphical example is presented, where the objects are images, see Fig. 5. The result shows that

Fig. 5 A graphical example: a) cover image (cover object), b) discrete wavelet transform and thresholding with $K \leq 15$ on cover image, c) plaintext (secret image), d) ciphertext image (ciphertext), and e) stego image (stego object).

the cover image and stego image are similar to the human eye. Nevertheless, the stego image has embedded into the ciphertext image. At this point, the plaintext (secret image) is transformed to the ciphertext, where pixels are ciphered by the cryptographic algorithm.

These data are reported due to the hardware architecture presents an optimal design, where only one *Adder* and one *Multiplier* are mainly used, increasing the latency and reducing hardware resources. Additionally, these modules are sequentially implemented for computing matrix multiplications, so operations are executed in sequential and iterative processes. This advantage provides low hardware resource consumption, low power consumption, greater latency (possibility of pipeline architecture) and low clock frequency. On the one hand, the latency depends on the sizes of the matrices H and F . On the other hand, by using an adequate design, the frequency reduction enables a lower power consumption [21]. Moreover, in the future, this design can support a pipeline architecture, see Fig. 4, which will improve the throughput for different communication lines, as in the automotive case, where there is a number of ECUs (Engine Control Units) up to 70 [22] in the contemporary

cars, whereas the number of sensors is forecasted to be as many as 200 per connected vehicle by 2020 [23].

7. Conclusion

The information age requires data protection through the network to guarantee security when data transmission takes place and even more when the transmitted data is confidential information. Steganography allows the shielding of information by hiding the data using a medium as a cover, in this case, an image. For avoiding attacks against confidentiality, cryptographic algorithm AES is used together with the Haar's discrete wavelet transform to hide the message. If these methods are implemented in software, they will cause slow processing due to the number of computational resources needed to execute their processing. Because of the operations of these algorithms and their mathematical complexity, it is recommended to implement it in hardware. The proposed implementation allows greater efficiency in hardware resources, allowing this design to be practical implementing it in a real system.

Acknowledgments

Authors express their gratitude to the Mexican National Council for Science and Technology (CONACYT) for financing this work through the Research Project 882 named "Development of Security Systems for Application on Automotive Industry in the Tlaxcala state".

References

- [1] T. Fujino, T. Kubota, and M. Shiozaki, "Tamper-resistant cryptographic hardware," *IEICE Electronics Express*, vol.14, no.2, 20162004, 2017, DOI: 10.1587/elex.14.20162004.
- [2] K.F. Rafat and M.J. Hussain, "Secure steganography for digital images," *International Journal of Advanced Computer Science and Applications*, vol.7, no.6, pp.45–59, 2016.
- [3] S. Baddap, K. Khomane, P. Deshmukh, and P. Shilpa, "Hardware implementation of LSB steganography for data security," *International Journal of Innovative Research in Advanced Engineering*, vol.2, no.3, pp.59–63, 2015.
- [4] M. Nagata, D. Fujimoto, N. Miura, N. Homma, Y.-I. Hayashi, and K. Sakiyama, "Protecting cryptographic integrated circuits with side-channel information," *IEICE Electronics Express*, vol.14, no.2, 20162005, 2017, DOI: 10.1587/elex.14.20162005.
- [5] V. Bánoci, G. Bugár, and D. Levický, "A novel method of image steganography in DWT domain," *Proc. 21st International Conference Radioelektronika*, 2011, DOI: 10.1109/RADIOELEK.2011.5936455.
- [6] NIST, "FIPS PUB 197: Advanced Encryption Standard (AES)," National Institute of Standards and Technology, 2001.
- [7] R. Oshana, ed., *DSP for Embedded and Real-Time Systems*, Newnes, Elsevier, Walham, 2012.
- [8] A.B. Lopez, K. Vatanparvar, A.P.D. Nath, S. Yang, S. Bhunia, and M.A.A. Faruque, "A security perspective on battery systems of the Internet of Things," *Journal of Hardware and Systems Security*, vol.1, no.2, pp.188–199, 2017, DOI: 10.1007/s41635-017-0007-0.
- [9] H.-T. Wu, Y.-T. Wu, and W.-W. Chang, "Biometric identification using JPEG200 compressed ECG signal," *IEICE Trans. Inf. & Syst.*, vol.E98-D, no.10, pp.1829–1837, Oct. 2015.
- [10] N. Hirata, T. Nozaki, and M. Kawamura, "Image watermarking method satisfying IHC by using PEG LDPC code," *IEICE Trans. Inf. & Syst.*, vol.E100-D, no.1, pp.13–23, Jan. 2017.
- [11] M.U. Umamaheswari, S. Sivasubramanian, and S. Pandiarajan, "Analysis of different steganographic algorithms for secured data hiding," *IJCSNS International Journal of Computer Science and Network Security*, vol.10, no.8, pp.154–160 (2010).
- [12] M. Weeks, *Digital Signal Processing Using MATLAB and Wavelets*, Infinity Science Press LLC, 2007, ISBN: 0-9778582-0-0.
- [13] X. Liao, H. Zhang, and M. Koshimura, "Reconstructing AES key schedule images with SAT and MaxSAT," *IEICE Trans. Inf. & Syst.*, vol.E99-D, no.1, pp.141–150, Jan. 2016.
- [14] I. Algreto-Badillo, C. Feregrino-Urbe, and R. Cumplido, "Design and implementation of an FPGA-based 1.452 Gbps non-pipelined AES architecture," *Computational Science and Its Applications, ICCSA 2006, Lecture Notes in Computer Science*, vol.3982, pp.456–465, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [15] B.A. Forouzan, *Cryptography and Network Security*, McGraw-Hill Education, 2007.
- [16] A.A. Jadhav, R.V. Babar, and M.S. Gaikwad, "Hardware implementation of digital watermarking system for video authentication using bit replacement algorithm & DWT," *International Journal on Emerging Trends in Technology*, vol.1, no.1, pp.111–114 (2014).
- [17] C.S. Maya and G. Sabarinath, "An optimized FPGA implementation of LSB replacement steganography using DWT," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol.2, Special Issue 1, pp.586–593, 2013.
- [18] A.F. Mahmood, N.A. Kanai, and S.S. Mohmmad, "An FPGA implementation of secured steganography communication system," *Tikrit Journal of Engineering Sciences*, vol.19, no.4, pp.14–23, 2012.
- [19] K.N. Pansare and A.K. Kureshi, "A review — FPGA implementation of different steganographic technique," *International Journal of Innovative Research in Science, Engineering and Technology*, vol.3, Special Issue 4, pp.377–381, 2014.
- [20] B.J. Mohd, T. Hayajneh, and A.N. Quttoum, "Wavelet-transform steganography: Algorithm and hardware implementation," *International Journal of Electronic Security and Digital Forensics*, vol.5, no.3/4, pp.241–256, 2013, DOI: 10.1504/IJESDF.2013.058657.
- [21] L.V. Bogdanov et al., "Approaches for reducing the power consumption in embedded systems," *Annual Journal of Electronics*, 2012.
- [22] N. Lu, N. Cheng, N. Zhang, X. Shen, and J.W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet of Things Journal*, vol.1, no.4, pp.289–299, 2014.
- [23] E. Schmidt, "Vehicle connectivity in intelligent transport systems: Today and future," *International Automotive Engineering Conference*, 2017.

Ignacio Algreto-Badillo received his Ph.D. degree in Computer Science from INAOE and, since 2017, he is a researcher of CONACYT. He has been involved in the design and development of digital systems, reconfigurable architectures, software radio platforms, cryptographic systems, FPGAs implementations, microcontrollers-based systems and hardware acceleration for specific applications.

Francisco Rubén Castillo-Soria received his M.S. in Telecommunications Engineering from IPN-ESIME in 2004 and his Ph.D. degree in Electronics and Telecommunications from the CICESE Research Center, Ensenada, in 2015. Since 2017, he has been a research-professor at Autonomous University of San Luis Potosí, SLP, México. His current research interests include SM, STF coding and MIMO-OFDM.

Kelsey A. Ramírez-Gutiérrez received her M.S. degree in 2010, and her Ph.D. in 2014 from the Mechanical and Electrical Engineering School of the National Polytechnic Institute of México. Her research interests are in the fields of biometrics, pattern recognition and information security.

Luis Morales-Rosales received his Engineering degree in Computer Systems from the Instituto Tecnológico de Colima, in 2001 and his Ph.D. in Computer Science in 2009 from the National Institute of Astro-physics, Optics and Electronics. His current research interests are applications of distributed systems and intelligent computing.

Alejandro Medina-Santiago received his M.S. and Ph.D. degree in Electrical Engineering from Center for Research and Advanced Studies of the National Polytechnic Institute. His line of research is VLSI integrated circuit design and intelligent systems design based on neural networks and fuzzy logic and Cybersecurity at software and hardware level.

Claudia Feregrino-Uribe is a researcher at the Computer Science Department at INAOE, Puebla, Mexico. Her research areas are Cryptography, Watermarking and Digital Systems Design. She received her B.Sc. in Computer Systems Engineering from Querétaro Institute of Technology, M.Sc. in Electrical Engineering with Telecommunications option from the CINVESTAV, Guadalajara and Ph.D. from Electronic Engineering in Digital Systems from Loughborough University in the United

Kingdom. She has published 100+ papers in scientific journals and international conferences, is associate editor for several international journals and has been involved in the organization or as a PC member for several conferences/workshops.