# INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación el 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

ESPECIALIDAD EN SISTEMAS EMBEBIDOS

# COMPARISON OF DIGITAL WATERMARKING METHODS FOR AN ID AUTHENTICATION SYSTEM

Tesina para obtener el grado de:

ESPECIALISTA EN SISTEMAS EMBEBIDOS

Presentan:
Francisco Javier Aguirre Ramos
Víctor Manuel Caamaño Salgado

Director:
Dr. Iván Esteban Villalón Turrubiates

San Pedro Tlaquepaque, Jalisco. 11 de Julio de 2018.

# Acknowledgments

The authors would like to thank to the Consejo Nacional de Ciencia y Tecnología (CONACYT) for trusting us.

To our parents, for their support in any decision we have ever made.

To our thesis advisor Dr. Iván Esteban Villalón Turrubiates for his continuous support and availability during the development of this research.

To Dr. Lorena Michele Brennan Bourdon for her contribution in the writing and revision processes of this document.

# Abstract

Current radio-frequency identification (RFID) card authentication systems are not secure enough to fight the latest and most novel hacking methods. Poor software implementations, outdated cryptography algorithms and faulty hardware are just some of the most common ways of exploiting these systems. Research in the field of image processing and cryptography suggests that an additional authentication layer based on digital watermarking could help improve the level of security of traditional RFID cards. Unfortunately, not all watermarking methods can be implemented in an embedded device, such as the one used for RFID card systems. The aim of this work is to provide a comparison among different digital watermarking techniques that can be used to add the extra security layer required by the basic RFID card authentication systems. In this work, two widely known methods proposed by Kang et al. in 2010 and 2003, were selected as the basis to create a comparison framework for their implementation in an embedded device. Important elements such as algorithm complexity and memory occupation were measured and analyzed in order to select the best candidate for an RFID card system. The method proposed by Kang et al. in 2010 represented the option with lowest algorithmic complexity and less memory footprint, indicating that this method is the most suitable for its implementation in an authentication system.

# Resumen

Los métodos de autenticación basados en tarjetas con tecnología RFID (del inglés, radio-frequency identification) no son lo suficientemente seguros para contrarrestar las técnicas de hackeo más novedosas y actuales. Implementaciones deficientes, algoritmos de criptografía obsoletos y errores de hardware son solo algunas de las formas más comunes para vulnerar este tipo de sistemas. Investigaciones en el área de procesamiento digital de imágenes y criptografía sugieren que una capa adicional de autenticación basada en marcas de agua digitales podría ayudar a incrementar el nivel de seguridad de las tarjetas RFID tradicionales. Desafortunadamente, no todos los métodos de marca de agua digital pueden ser implementados en un sistema embebido como el usado en los sistemas de tarjetas RFID. El objetivo de este trabajo en proveer una comparación entre diversas técnicas de marcas de agua digital que pueden ser usadas para proveer una capa de seguridad extra a los sistemas de autenticación basados en RFID. En este trabajo, dos métodos ampliamente conocidos propuestos por Kang y Cols. en el 2010 y 2003, fueron seleccionados como base para crear un marco de comparación para su implementación en un sistema embebido. Elementos importantes tales como complejidad algorítmica y ocupación de memoria fueron medidos y analizados para elegir el mejor candidato para un sistema de tarjetas RFID. El método propuesto por Kang y Cols. en 2010 representó la opción con la complejidad algorítmica más baja y menor ocupación de memoria, indicando que este método es el más apto para su implementación en un sistema de autenticación como el deseado.

# List of figures

# List of tables

# Abbreviations and acronyms

| | |
|---|---|
| **DCT** | *Discrete Cosine Transform* |
| **DFT** | *Discrete Fourier Transform* |
| **DWT** | *Discrete Wavelet Transform* |
| **FFT** | *Fast Fourier Transform* |
| **FPGA** | *Field-Programmable Gate Array* |
| **GPU** | *Graphics Processing Unit* |
| **HVS** | *Human Vision System* |
| **ID** | *Identification* |
| **IDFT** | *Inverse Discrete Fourier Transform* |
| **ILPM** | *Inverse Log-Polar Mapping* |
| **PC** | *Personal Computer* |
| **RAM** | *Random Access Memory* |
| **RFID** | *Radio-Frequency Identification* |
| **ROM** | *Read-Only Memory* |
| **RSTC** | *Rotation, Scaling, Translation, and Cropping* |
| **ULPM** | *Uniform Log-Polar Mapping* |

# Content

# 1. Introduction

Information security plays a central role in any company, regardless of its size, whether it is a large corporation with thousands of employees or just a small company with limited resources. What is important is to be able to guarantee the secrecy of those elements classified as sensitive information.

One of the first barriers against any intrusion is access control. In this field, contactless technology, which is the group of technologies used to identify objects by means of radio frequency signals, has gained increased popularity [1], [2]. There are a number alternatives that make use of this technology; Radio Frequency Identification (RFID) cards are the most popular solutions since they provide through a unique identification tag, a reliable and simple way to implement access control that can be applied to any organization [3].

Unfortunately, security branches have become more common nowadays [4]. As a result, authentication systems must increase their robustness in order to provide the required security levels to match the latest technological developments. In this sense, RFID card based systems can become a highly insecure solution if they are not implemented carefully [5]. Some of these systems appear to be stalled in time, consequently, they can be effortlessly branched by using basic equipment that can be obtained through the internet. The digital black market is no longer required, all the technology that threatens these systems can be acquired directly through popular sites [6], such as eBay or Amazon [7], [8].

It is fair to say that basic RFID card authentication systems have been overpassed and are no longer reliable. However, despite of their faults, these kinds of systems remain popular around the world. They are used in a large number of companies to provide different levels of access control [9]. Given the current circumstances it is important to shield these traditional methods by combining them with novel techniques with increased security. These techniques must be capable of providing an additional layer of security without further complications or additional high costs.

One of the options that has become more popular in the last years is digital watermarking [10]. It is classified as a cryptography technique and has many applications in the new digital world. The main idea behind them is to embed information into a digital media; this information is undetectable by the Human Vision System (HVS) and, ideally, by advanced artificial methods as well [10]. The information can be embedded into different parts of the multimedia, depending on the nature of the media, trying to keep such information unchanged. In this sense, digital watermarking applications are vast, they can be basic, such as embedding subtitles into a movie (video watermarking) [11] or more complex, for example, linking an author to its digital creation (digital fingerprinting) [12], and that is just the tip of the iceberg, new applications are being created every day.

As with many cryptography techniques, digital watermarking can be used to improve the level of security of an existing system. In this case, the traditional RFID card authentication security can be dramatically improved by introducing a new security layer based on digital watermarking. This extra level will involve adding information to the card by means of a digital watermarking technique. The resulting watermark must be capable of resisting specific attacks associated to the authentication processes, such as printing and scanning, which are necessary during the creation of the enhanced RFID identification card and its validation by a reader.

The entire validation process of an RFID identification card, as the industry requires, has to be part of an embedded system. This includes the creation and the validation of the digital watermark; therefore, the selected technique must be compatible with its implementation in an embedded system [13].

## 1.1. General Objective

The main objective of this work is to provide a comparison among different digital watermarking techniques that can be used to add the extra security layer required by the basic RFID card authentication systems. This comparison process will be carried out by means of objective metrics obtained as a result of the partial implementation of key elements intrinsic to these methods.

# 2. Background

Since the origin of the digital watermarking concept, several computational techniques to improve resiliency of watermarks over digital mediums, such as videos, texts, and images, have been thoroughly explored. These techniques have been centered more on the quality of the watermark than in the amount of computational resources, like the memory usage and processing time that these methods require to obtain the referred quality. For this reason, a set of digital watermarking methods are described emphasizing the characteristics of their resulting watermarks along with the type and amount of computational resources used during its processing.

The work done by Guo et al. entitled "*A color image watermarking algorithm resistant to print-scan*" [14] presents a blind watermarking method resistant to print-scan attacks. The proposed technique consists of dividing the image into different blocks and calculating the image complexity of each block in order to determine if the watermark can be located in this particular partition. Once it has been determined that the portion is a candidate for watermark location, the information is embedded into the mid-frequency coefficients of the Discrete Fourier Transform (DFT).

For the purpose of the present thesis, one of the disadvantages detected in this method is that high quality printing and scanning processes are needed in order to maintain the watermark resistant. These aforementioned processes are seldom supported by embedded devices, which makes this method unaffordable.

As a novel watermarking technique, Keskinarkaus et al. [15] propose a multibit watermarking system based on coordinate points in the image. Using these coordinates, the watermark can be retrieved even after the printing and scanning attacks, however, the complexity of the involved operations shall be evaluated in order to determine the feasibility of its implementation in an embedded device.

Alternatively, Kang et al. [16] presents another watermarking algorithm resistant to attacks coming from geometric distortion and print-scan process. The main intention of this work is to propose a watermark method based on uniform log-polar mapping (ULPM) instead of the inverse log-polar mapping (ILPM) commonly used in prior investigations. The major advantage of using ULPM in the proposed algorithm is that it gives the effect that the embedded information happens in Fourier log polar domain when the embedding process occurs in fact, in Fourier Cartesian domain, this way not only the interpolation and the interface distortions are removed completely but the space to embed information is increased significantly. Additionally, the proposed watermark technique also provides a secret pattern of multi-points where the information is embedded based on a secret key which enhance the undetectability of the hidden information in order to avoid any intrusion.

A final watermarking technique also described by Kang et al. in the article entitled *"A DWT-DFT Composite Watermarking Scheme Robust to Both Affine Transform and JPEG Compression"*[17]. In this work, the authors propose a watermarking scheme resistant to affine transforms and JPEG compression. The algorithm described in the article, presents a blind watermarking technique, which combines Discrete Wavelet Transform (DWT) and Discrete Fourier Transform (DFT) as mechanisms to embed information within the image.

The resulting watermark in this method is accomplished by introducing the watermark information in low-frequency subband ($LL_4$) of the DWT spectrum and modulating a secret template in the mid-frequency magnitudes of the DFT domain.

The strength of this technique relies on the high level of imperceptibility of the immersed watermark and at the same time, its capability to provide the required redundancy to retrieve the watermark after the scan-printing attacks.

# 3. Theoretical Framework

## 3.1. Digital watermarking

Digital watermarking consists of embedding information into a media with different purposes and it can involve simple applications like enriched media content or complex applications, such as secret communications. Embedding a watermark into media has some challenges and limitations, since the existing techniques are designed to assure one or more of the following characteristics [18]:

- **Robustness:** It refers to the resilience of the watermark against attacks. Such attacks can be intentional or part of a given process; for this specific application, it is fundamental to be resilient against scanning/printing attacks.

- **Undetectability:** It is related to the impossibility of proving the existence of additional information within the media. It is restricted to just detection, not necessarily to recover the information. For some specific applications, such as covered communications, it is fundamental to avoid the detection of hidden data in the media.

- **Invisibility:** It refers to the inability of the HVS to detect the watermark and also to the level of degradation of the media. Any additional information in the media will introduce a reduction in quality; the characteristics of the watermark and the used technique can reduce this disturbance to a minimum.

- **Capacity:** It is related to the quantity of encoded information that a watermark may contain. This category is divided into two classes [19]:

    - ***Zero-bit:*** In this class, the data payload of the watermark is theoretically zero-bit. It is used for those systems where only one possible watermark can exist. The systems that utilize this watermark class are equipped solely with a mechanism capable to detect the presence or absence of the watermark.

- ***Multi-bit:*** In this class, also known as *non-zero-bit* class, the data payload of the watermark equals *N* bits. It is used for those systems where multiple watermarks can exist. Unlike zero-bit class, the systems that make use of multi-bit watermarks are provided with two the mechanisms, the first capable of detecting the presence of the watermark, and the second in charge of decoding the embedded information of the watermark.

These characteristics are not required in every application and a certain level of compromise between them is allowed. On the other hand, depending on the application, sometimes large amounts of information have to be embedded and this is translated as a high level of capacity. The embedding capacity cannot be increased indefinitely; there is a trade-off between capacity, undetectability, and robustness (Figure 3-1). The selection of a good watermarking technique shall be based on these three characteristics [18].

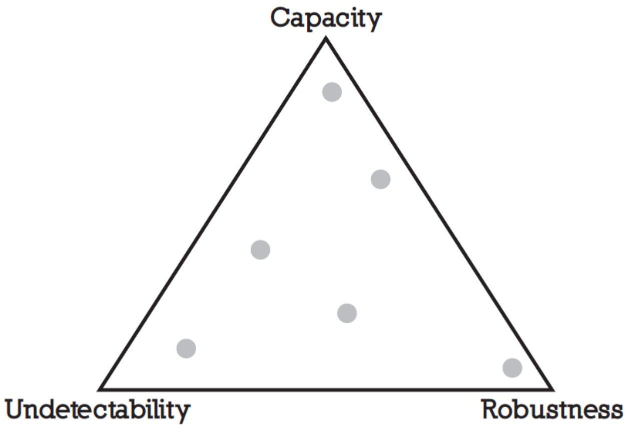

*Figure 3-1. Trade-off triangle between capacity, undetectability, and robustness.*

There are many ways to classify watermarking algorithms, one of the most popular classification is based on their reversibility:

**Reversible:** This method is capable of retrieving both the embedded information and the original media where the information was immersed. This means that the process used to embed information is completely reversible.

**Non-reversible:** In this method, when the embedded information is retrieved, the original media is affected. As a result, the media cannot be recovered in its original form. In other words, this method involves processes that cannot be reverted.

Another classification of the watermarking techniques is according to the embedding domain. This classification is of special interest for the present work since it determines the algorithm that will be used. There are two types in this category:

### 3.1.1 Spatial Domain

Watermarking technique under spatial domain consists of selecting random areas over the image to be processed and modifying the belonging pixels of these areas based on a secret signature utilized by the author to mark the image.

Spatial domain considers three factors for its design: the information related to the signature, the random pattern where the information will be embedded, and the type of mask used for the image. Considering these three factors, the resulting mathematical model for this algorithm is as follows:

$$\hat{y}_{\mathrm{i,j}} = y_{i,j} + \propto I \tag{3-1}$$

Where $y_{i,j}$ corresponds to the original intensity level of the image at pixel $(i, j)$, $\hat{y}_{\mathrm{i,j}}$ is the resulting marked image and $\propto I$ represents the embedded information of the marked image in terms of the intensity level variation over each pixel.

In order to obtain the watermark included in the image, two secret keys are needed, the first containing the random pattern where the embedded information is located and the second, with the information of the watermark [20].

### 3.1.2 Frequency Domain

In frequency domain, the base image is first separated into different sections. Over each section DCT, DFT or any other discrete transform algorithm, are applied. The resulting coefficients derived from discrete transform methods are then combined with the desired watermark information. In frequency domain, as well as in spatial domain, the sections of the image are selected in a random manner. The mathematical formula that represents this method is the following is described in (3-2).

$$\hat{C}_{i,j} = C_{i,j}\,(1 + \beta I) \tag{3-2}$$

In this case, $\hat{C}_{i,j}$ represents the resulting coefficients of watermark image at pixel $(i, j)$, $C_{i,j}$ are the discrete transform coefficients (DCT, DFT, etc.) of the selected section in the image, and $\beta I$ determines the factor of modulated watermark based on the entropy of the selected sections in the image and the data that will be embedded [20].

# 4. Methodology

As it was mentioned earlier, the main objective of this work is to evaluate certain digital watermarking methods and the feasibility of their implementation in an embedded device. Many different aspects can be considered during this evaluation and several methods can be evaluated. However, our specific application reduces the number of watermarking methods of interest to those that are designed specifically to endure the print-scanning processes, which involve a combination of different attacks, such as rotation, scaling, translations, and cropping (RSTC). Also, a modification in the pixel values is introduced during the printing and scanning processes.

In this sense, it was decided to restrict the field of interest to those methods that mention in their description the resilience against RSTC attacks. Among the strongest candidates, the proposal by Kang, et al. [16] is compared against a previous work of the same authors [17], showing a significant improvement in terms of robustness against the RSTC attacks. These two proposals are considered ideal candidates for their implementation in an embedded device.

It is also important to restrict the main features to evaluate the implementation in an embedded device. The most common features are related to memory occupation (Random Access Memory RAM, and Read-Only Memory ROM) and algorithm complexity. However, these aspects are too broad and general to be evaluated in the application of interest. The ID card authentication problem based on digital watermarking requires the implementation of a reliable but lightweight method; these kinds of systems are usually implemented in small devices based on simple processors with very limited resources.

Low complexity processors are a synonym of low price products but also of optimized software applications. Special attention shall be placed in parameters, such as ROM occupation in terms of program footprint and the complexity of the operations involved in the algorithm. These two metrics are the basis of our proposed evaluation and obtaining these metrics is not an easy task. The straightforward solution would be to implement the methods and measure the complete system. Unfortunately, this is not a feasible solution.

Implementing even one of these two methods in an embedded device or even in a normal PC would require a high effort and a series of development and testing cycles. Also, the purpose of this investigation is not the construction of an ID card authentication system, but to provide a framework of reference to determine which method would be the most suitable to these authentication systems. Therefore, it was decided to analyze the methods by partially implementing them, therefore, just certain elements of each method will be implemented and measured. The partial implementation will provide a starting point to obtain a good approximation of the total complexity of the method in terms of program footprint and complexity.

After analyzing both methods [16] and [17], it was decided to evaluate the following features:

- Watermark embedding complexity: This is the first stage and it involves the preparation of the information that will be the watermark. A series of operations and transformations are required during this stage. Most of them are related to transform calculations and to obtain additional information for embedding.

- Watermark extraction complexity: The second stage of both methods is the most important. While the first stage can be performed offline in a larger system, the watermarking extraction shall be performed in real time. This means that the restrictions of resources for the second stage are tighter. During this stage, the inverse process is carried out to locate the watermark, retrieve it, and decode it.

The aforementioned features represent the core elements for both methods and basically what defines them. Therefore, their implementation cannot be completed without them. It is important to mention that this evaluation considers the specific application for ID card authentication; different applications might require additional measures. Nonetheless, both ROM occupation and operation complexity are basic metrics used to decide over the implementation feasibility in any embedded device.

All implementations, tests, and evaluations were implemented using GNU Octave 4.0.0 [21] and C/C++. The testing equipment characteristics are: OS Microsoft Windows 7 Enterprise, Intel®

Core™ i7 CPU@2.70GHz with 16GB on RAM. It was used to validate the complexity of the evaluated methods.

# 5. Results

Results of this study are divided in two sections, the first is related to algorithmic complexity and the second is focused on ROM occupation. The calculation of the algorithmic complexity was performed by means of a step by step analysis of both methods; the complexity is presented in big-O [22] notation to include the worst case scenario. On the other hand, the ROM occupation is presented without units, and expressed as a comparison in percentages; this way, it is easier to know which method provides the smaller memory footprint.

While ROM occupation is important for every embedded system, the algorithmic analysis included in this chapter could be classified as more relevant for the present work. Given the real-time constraints of an ID authentication system, an important delay in the detection and decoding of the watermark would translate into a low performance system which is not suitable for the various environments where these systems are present.

The processes of embedding and extraction of watermark information can be considered as inverse operations, which represents that the algorithmic complexity of both procedures is theoretically the same. Embedding and extraction steps are applied at different stages of the entire watermarking process; the embedding step is performed at the beginning and can be executed offline while extraction is realized at the end of the process and shall be executed at runtime. As consequence of the timing constraints presented in the watermark extraction process, the analysis of its algorithmic complexity becomes more relevant than for the embedding step. For this reason, the results of the present thesis are focused in the extraction part of each watermarking method.

## 5.1. Algorithmic Complexity

The most algorithmically relevant steps of the extraction process in [16] are represented in Figure 5-1, we have four major steps:

1. Fast Fourier transform over the image. This is the first step and it is fundamental in order to start to search the tracking pattern. Given that the operation is performed over the full $N*M$ image, the computational complexity of the 2D FFT is delimited by $O(M * N * \log(M * N))$.

2. Uniform Log-Polar Mapping (ULPM) calculation. As a second step, it is necessary to map the coefficients using the ULPM technique. This is basically to map each DFT coefficient $F(l_1, l_2)$ according to (5-1) and (5-3). The importance of this step is that the ULPM operations are applied to all coefficients and they can be delimited by $O(M * N)$.

$$l_1 = floor\left(\log_a \frac{r}{R}\right) + \frac{M}{2} \tag{5-1}$$

$$a = 2^{1/M}, r = \sqrt{u^2 + v^2}, R = 0.2 \tag{5-2}$$

$$l_2 = floor\left(\frac{N \times \theta}{\pi}\right) \tag{5-3}$$

$$\theta = \tan^{-1}(u/v) \tag{5-4}$$

3. Cross correlation. This step is performed by means of a procedure proposed by the authors. Along different operations, the calculation of the IDFT and a pixel average calculation is noticed. This process has a similar complexity than the normal DFT calculation plus a set of operations over all coefficients; therefore, we can delimit this step using $O(M * N * \log(M * N)) + O(M * N)$. Again, only the most computationally complex operations are analyzed, leaving basic operations constant.

4. Watermark decoding. In this stage the original information is decoded from the retrieved watermark. The process is carried out over the original coded sequence which is of size $\frac{M}{2} \times \frac{N}{2}$; the algorithmic complexity can be limited by $O\left(\frac{M}{2} * \frac{N}{2}\right)$.

Finally, it can be determined that the algorithmic complexity of the extraction method described in [16] is given by (5-5) which can be reduced to (5-9) if we assume a square image.

$$O\left(M * N * \log(M * N) + M * N + \ M * N * \log(M * N) + M * N + \frac{M}{2} * \frac{N}{2}\right) \tag{5-5}$$

$$O\left(N^2 * \log(N^2) + N^2 + \ N^2 * \log(N^2) + N^2 + \frac{N^2}{4}\right) \tag{5-6}$$

$$O\left(N^2 * \left(\log(N^2) + 1 + \log(N^2) + 1 + \frac{1}{4}\right)\right) \tag{5-7}$$

$$O\big(N^2 * (\log(N^4) + c)\big) \tag{5-8}$$

$$O(N^2 * \log(N^4) + N^2) \tag{5-9}$$

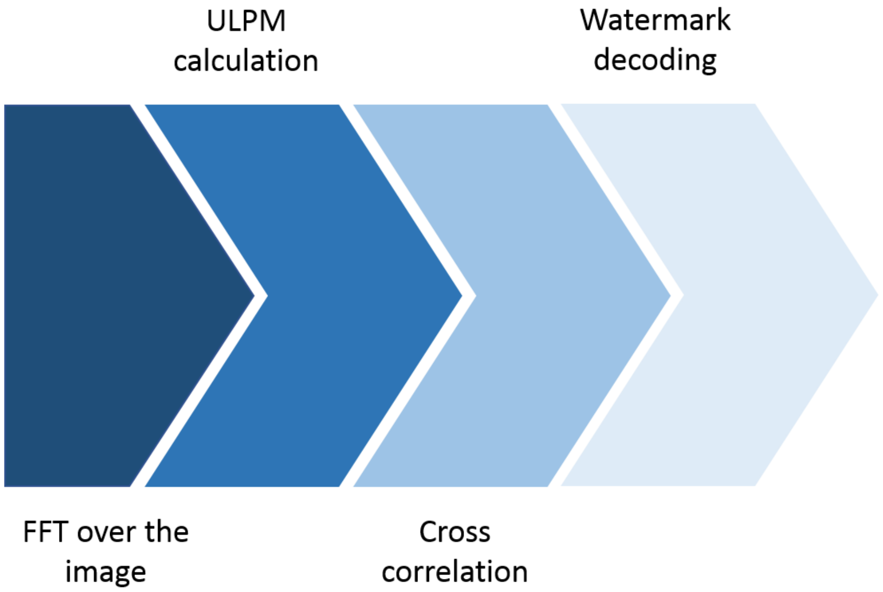

*Figure 5-1. Extraction method as described in [16].*

In comparison, the method described in [17] makes use of two different transforms, Wavelets and Fourier. Both transforms have a similar complexity under the constraints specified in the method, with slight constant variations that can be obviated.

Figure 5-2 shows the main stages of the method in [17]; again, these stages are the most computationally demanding, therefore, they are the subject of interest of this evaluation. Once again, each stage is analyzed for complexity using big-O notation; in this specific method, each stage involves a high number of operations, some of them are not necessary under certain conditions. However, we are analyzing for the worst case, therefore, all operations are considered. The three main stages of the method are:

1. Discrete Wavelet Transform calculation. As a first stage, the DWT is calculated over the full image of size $M * N$. Having the same computational complexity than the DFT, we have $O(M * N * \log(M * N))$ for this stage.

DWT
Resync
Retrieval

*Figure 5-2. Main stages of the extraction method as described in [17].*

2. Resynchronization – Template Detection. If the image suffered an affine transformation (attack) it is required to retrieve the template from the DFT domain. The first step is to apply a Bartlett window [23] to all the image, this operation is of the order $O(M * N)$. The next step is to apply a DFT to the full image; this procedure is applied over a higher resolution image (double the size). The algorithmic complexity would be limited by $O(2 * M * N * \log(M * N))$. After the DFT calculation the image is processed looking for local peaks, this process has a computational complexity of $O(2 * M * N)$. The local peaks are processed adding $O(2N)$ to the complexity. Solving two linear equations is required for the next step; it is known that solving a linear equation of $n$ variables has a complexity of $O(n^3)$, and for our specific case it would be $O(2^3)$.

3. Extraction and decoding. This process requires of a semi-brute force method that in the worst case performs around 256 comparisons over the image with the DWT calculation involved. This means $O(256 * M * N * \log(M * N))$; after this process ends, a final scanning over the image is required adding additional $O(M * N)$ to the complexity.

The full complexity of the three extraction stages is resumed in (5-10), and it can be reduced to (5-14) by supposing a $N * N$ image.

$$O(M * N * \log(M * N) + M * N + 2 * M * N * \log(M * N) + 2 * M * N + 2 * N + 2^3 + 256 * M * N * \log(M * N) + N * M) \tag{5-10}$$

$$O(N^2 * \log(N^2) + N^2 + 2 * N^2 * \log(N^2) + 2 * N^2 + 2 * N + 2^3 + 256 * N^2 * \log(N^2) + N^2) \tag{5-11}$$

$$O\left(N^2\left(\frac{2}{N} + \frac{2^3}{N^2} + 259 * \log(N^2) + 4\right)\right) \tag{5-12}$$

$$O\left(N^2\left(\frac{2}{N} + \frac{2^3}{N^2} + \log(N^{518}) + c\right)\right) \tag{5-13}$$

$$O(2 * N + 2^3 + N^2 * \log(N^{518}) + N^2) \tag{5-14}$$

It is obvious that the method in [17] is considerably more complex than [16]. In fact, it is so complex that its implementation in an embedded device would require a specialized architecture based on a field-programmable gate array (FPGA) device or a parallel implementation based on graphics processing units (GPUs).

## 5.2. ROM Occupation

Main elements of each method were implemented using C/C++ programming language. As it was expected, ROM occupation in both methods is similar. Many of the operations in the most demanding parts of each method are common; therefore, they are optimized by the compiler into one set of instructions. The ROM values are presented in Table 5-1 and Table 5-2, the occupation increases in about 0.6% from method [16] to [17].

*Table 5-1. Results of ROM occupation by each method [16] in bytes.*

| text | data | bss | dec | hex |
|---|---|---|---|---|
| 21888 | 9284 | 12988 | 44160 | ac80 |

*Table 5-2. Results of ROM occupation for method [17] in bytes.*

| text | data | bss | dec | hex |
|---|---|---|---|---|
| 22036 | 9284 | 12988 | 44308 | ad14 |

Even when the ROM occupation does not represent major changes in this case; it is important to mention that one common way of improving execution times is to declare inline functions. This would significantly increase the ROM occupation of method [17].

# 6. Conclusions and Future Work

The main contributions presented in this work are summarized below:

1. A comparison between two watermarking schemes resistant to RSTC attacks was performed. Such attacks are derived from print-scanning methods inherent to the ID card authentication process. The first scheme [16] evaluated under the present work makes use of the discrete Fourier transform while the second scheme [17] combines two different transforms: Fourier and Wavelet. For both watermarking schemes their computational complexities were analyzed giving as a result that the first watermarking method [16] is composed by a lower algorithmic complexity defined in (6-1).

$$O(N^2 * \log(N^4) + N^2) \tag{6-1}$$

   In comparison with the complexity of the second watermarking method defined in (6-2).

$$O(2 * N + 2^3 + N^2 * \log(N^{518}) + N^2) \tag{6-2}$$

2. Besides the comparison of the computational complexity of the selected watermarking methods, the ROM occupation of each watermark scheme was evaluated as well. This evaluation was accomplished by measuring the amount of memory required by the set of C/C++ program subroutines needed to represent, in an accurate manner, the computational complexity previously obtained. In this study, the resulting ROM occupation values revealed that the second watermarking method [17] required slightly higher amount of memory than the first watermarking scheme [16]. This was expected, given that most of the executed procedures can be optimized in space during compilation; however, this is not related to its computation complexity, which does not decrease by any means.

3. Based on ROM occupation and computational complexity, resulting values concluded that the watermarking method proposed by Kang et al. in [16] represents the most suitable solution to be implemented within an embedded device not only because it reflects an slight

improvement in memory consumption, but also because it represents the simplest watermarking algorithm with considerably reduced processing time, which is limited in most of the embedded systems.

The future work planned for the present thesis aims to implement the selected watermarking method in an actual embedded system, e.g. a RFID card reader, evaluating its entire performance versus the quality of the watermarked image in terms of undetectability and robustness.

# References

[1] S. Ortiz, “Is near-field communication close to success?,” *Computer*, vol. 39, no. 3, pp. 18–20, Mar. 2006.

[2] “Contactless Technology Overview,” *MWR InfoSecurity*. [Online]. Available: https://www.mwrinfosecurity.com/our-thinking/contactless-technology-overview/. [Accessed: 08-Jun-2018].

[3] “4 Companies that Adopted RFID Access Control,” *RFID Insider*, 04-Jun-2015. .

[4] “SEC.gov | The Need for Greater Focus on the Cybersecurity Challenges Facing Small and Midsize Businesses.” [Online]. Available: https://www.sec.gov/news/statement/cybersecurity-challenges-for-small-midsize-businesses.html. [Accessed: 30-May-2018].

[5] “Top 10 RFID Security Concerns and Threats | securitywing.” .

[6] D. Ryan, “Watch How Easy It Is for Your RFID Card to be Cloned By Hackers [Video].” [Online]. Available: https://insights.identicard.com/blog/watch-how-easy-it-is-for-your-rfid-card-to-be-cloned-by-hackers-video. [Accessed: 30-May-2018].

[7] “eBay.com: NFC IC ID Copier Duplicator RFID Reader Writer Access Key Duplicator 13.56MHz,” *eBay*. [Online]. Available: http://r.ebay.com/us55Ah. [Accessed: 08-Jun-2018].

[8] “Amazon.com: FONGWAH UHF RFID USB Reader/Writer,” *Amazon.com*. [Online]. Available: http://a.co/3INAQCK. [Accessed: 08-Jun-2018].

[9] *What is RFID Used for in the Real World? | RFIDinsider*. 2013.

[10] C. Honsinger, “Digital Watermarking,” *J. Electron. Imaging*, vol. 11, no. 3, p. 414, Jul. 2002.

[11] K. Liao, S. Lian, Z. Guo, and J. Wang, “Efficient information hiding in H.264/AVC video coding,” *Telecommun. Syst.*, vol. 49, no. 2, pp. 261–269, Feb. 2012.

[12] L. Chun-Shien, *Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property*. Idea Group Inc (IGI), 2004.

[13] M. D. P. Emilio, *Embedded Systems Design for High-Speed Data Acquisition and Control*. Springer International Publishing, 2015.

[14] C. Guo, G. Xu, X. Niu, Y. Yang, and Y. Li, “A color image watermarking algorithm resistant to print-scan,” in *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, 2010, pp. 518–521.

[15] A. Keskinarkaus, A. Pramila, and T. Seppänen, “Image watermarking with feature point based synchronization robust to print–scan attack,” *J. Vis. Commun. Image Represent.*, vol. 23, no. 3, pp. 507–515, Apr. 2012.

[16] X. Kang, J. Huang, and W. Zeng, “Efficient General Print-Scanning Resilient Data Hiding Based on Uniform Log-Polar Mapping,” *IEEE Trans. Inf. Forensics Secur.*, vol. 5, no. 1, pp. 1–12, Mar. 2010.

[17] X. Kang, J. Huang, Y. Q. Shi, and Y. Lin, “A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 776–786, Aug. 2003.

[18] F. Aguirre-Ramos, “Error Concealment Method based on Data Hiding for HEVC,” Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Tonantzintla, Puebla, 2013.

[19] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography*. Morgan Kaufmann, 2007.

[20] W. N. Cheung, “Digital image watermarking in spatial and transform domains,” in *2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No.00CH37119)*, 2000, vol. 3, pp. 374–378 vol.3.

[21] “GNU Octave.” [Online]. Available: https://www.gnu.org/software/octave/. [Accessed: 11-Jul-2018].

[22] D. E. Knuth, “Big Omicron and Big Omega and Big Theta,” *SIGACT News*, vol. 8, no. 2, pp. 18–24, Apr. 1976.

[23] S. W. Smith, *The scientist and engineer’s guide to digital signal processing*. San Diego, Calif.: California Technical Pub., 1999.