Invited Paper: Implementing digital data hiding algorithms in reconfigurable hardware - Experiences on teaching and research

Rene Cumplido and
Claudia Feregrino-Uribe
Computer Science Department,
National Institute for Astrophysics, Optics and Electronics, INAOE
Tonanzintla, Puebla, Mexico
Email: rcumplido@inaoep.mx

Jose Juan Garcia-Hernandez
Information Technologies Laboratory, LTI
Center for Research and Advanced Studies, CINVESTAV
Tamaulipas, Mexico
Email: jjuan@tamps.cinvestav.mx

***Abstract*—Digital data hiding algorithms have recently received attention of the research community as an alternative to fight the piracy problem in the Internet era. Several data hiding applications, such as broadcasting monitoring and live performance watermarking, require a real-time multi-channel behavior, also new applications are constantly pushing the limits of available computing systems. This motivates the research on custom architectures, being reconfigurable logic a good option to implement such processing systems. This paper introduces the field of digital data hiding, including a brief review on hardware based architectures and a discussion of the challenges of implementing custom architectures for this type of applications. Also, the authors' experience in teaching and research on hardware architectures for digital data hiding algorithms at the MSc Program on Computer Science at INAOE is discussed.**

I. Introduction

The fast growth of the internet has increased the easy reproduction and retransmission of multimedia contents and, as a consequence, both legal and unauthorized data manipulation has also grown. Data hiding systems have emerged as a solution against the piracy problem, in which an imperceptible and statistically undetectable signature to multimedia content is added [1]. That signature must completely characterize the person who embedded it and in order to be used to prove the intellectual property of a digital media, any unauthorized removing or manipulation of the signature must render the digital media useless.

Several data hiding applications, such as broadcasting monitoring and live performance watermarking, require a real-time multi-channel behavior [2], [3]. While Digital Signal Processors (DSP) have been used for implementing these schemes achieving real-time performance for audio signal processing [4], [5], those implementations do not exploit the possible parallelism of several watermarking algorithms.

Even though microprocessor performances increase significantly every year, new applications are always pushing the limits of available computing systems. There is always the need of having more efficient processing systems in terms of performance, cost and power consumption. An alternative to meet these requirements has been the development of custom processing systems based on coprocessing units or full stand-alone custom architectures, both implemented in ASICs. More recently, FPGAs have emerged as an alternative to implement such processing systems. However, the set of skills needed to design systems based on programmable hardware such as FPGAs is broad, engineers need to have a good knowledge of several fields such as: computer architecture, digital design, software programing languages, and hardware description languages. Additionally, a good understanding of the application domain is desired.

The purpose of this paper is two-fold; (1) We introduce the reader to the field of digital data hiding, including a brief review on hardware based architectures and a discussion of the challenges of implementing architectures for this type of applications. (2) We describe the approach used to teach master students on computer science the skills needed to design custom processing architectures at the Computer Science department at INAOE. In particular, we focus on our experience in teaching and research on hardware architectures for digital data hiding.

The rest of the paper is organized as follows: Section II presents a background about digital data hiding. Section III describes the typical taxonomy of hardware architectures for digital data hiding schemes. Our experience in teaching and research about this subject is described in Section IV. Finally, Section V concludes this paper.

II. Data Hiding in Digital Media

Digital data hiding can be defined as the *art* of embedding information in a digital media, the *host signal*, without introducing any perceptual distortion. Digital watermarking and steganography are the main concepts devoted to exploit the hidden data for different purposes. On one hand, watermarking has been considered as one of the techniques with capacity to solve problems such as unauthorized copying and distribution of digital materials [6]. In order to be considered suitable for practical applications, watermarking algorithms must satisfy some requirements, such as imperceptibility of the embedded signal (watermark) and robustness to some common

intentional and/or non intentional attacks. These requirements, mainly robustness, limit the payload to a few bits of hidden data. On the other hand, steganography is a kind of secret communication using digital multimedia as the communication channel, therefore, the main demand is for both a high payload and high perceptual transparency. Contrary to watermarking algorithms, in steganographic systems the robustness is not an important issue [1]. An ideal steganographic scheme should have a large embedding capacity and excellent perceptual transparency [7]. However, in this paper watermarking and data hiding are used indistinctly. From the mid-nineties, data hiding has received increasing interest from the scientific community. This attention was motivated by the fast grown of the Internet and, in consequence, by the wide spread of digital media distribution, which revealed the need of protection for intellectual property rights of digital contents. In this scenery, watermarking seemed to be a potential solution [8].

A. Applications

Some applications of data hiding in digital contents are described next:

- Ownership demonstration. In this case, the watermark is embedded in order to prove unambiguously the intellectual ownership. The robustness is the main feature since the opponents are supposed to be aware of the watermark existence and a wide range of attacks has to be considered [9].
- Fingerprinting. In this case, the watermark is embedded in order to prove unambiguously the buyer. A different watermark, called fingerprint, is inserted in each copy being distributed. In this way, if an unauthorized copy of the protected work is found, the owner of the copyright can retrieve the identity of the buyer that illegitimately distributed the content [10]–[12].
- Copy control. Here, the embedded watermark signal contains rules of usage and copying [13]. In that scenery the recording device scans the digital data stream for an existing watermark and enables or disables the recording action for a specific movie or stream.
- Broadcast Monitoring. Data hiding systems can be used for tracking and monitoring advertisement activities on broadcasting channels and for examining advertisement deals. A label, which identifies a multimedia content to be broadcasted, can be hidden within the content itself to systematically detect when it is on air and to provide immediate reporting [14], [15].
- Covert communications. The main goal is to hide a large message into the digital medias keeping the undetectability of the presence of the message [10], [12].
- Authentication. In this case, the watermark, which is modified together with the host signal, reveals when it has been tampered, even after small changes, so that modifications on the watermarked content can be detected [10], [12], [16]. These schemes are said to use fragile watermarks.

B. Requeriments

The development of a data hiding system, depending on the particular application, involves several trade offs between unlike requirements. Digital watermarks must fulfill the following, often contradictory, requirements: [17]

- Robustness. Robustness means the resistance ability of the watermark information to changes and modifications made to the original file. Above all, commonly used operations such as lossy compression (JPEG, MPEG) should not destroy the digital watermark [18]. Robustness means resistance to common operations applied in the imaging, motion picture, or audio field [19].
- Imperceptibility. The watermarked signal should be perceptually indistinguishable from the original one. In other words, the embedding process has not to downgrade the quality of the digital content. Perceptual analysis has been widely exploited to find the optimal embedding domains and perceptual masks that maximize the power of the watermark without impairing the perceived quality of the content [8].
- Payload. It is the amount of information that the watermark signal is able to represent. This requirement is related with the application.
- Security. It may not be possible without knowledge of the procedure and the secret key to remove the watermark or to make it illegible. Security is usually related with the ability of a data hiding scheme to protect some secret parameter, so that an attacker can not use it to access the watermark contents [20].

In every watermarking scheme there exists a trade off between the robustness, maximum allowable distortion to the host, and the payload. Some watermarking schemes that use perceptually significant parts of the host to embed data in a robust manner have been proposed [21], [22]. In more recent works, perceptual models have been used for adaptive watermarking [23], [24].

C. Classification Based on Watermarking Domain

The data hiding schemes can be classified based on the watermarking domain.

- The earlier works in watermarking, such as least significant bit (LSB) replacement scheme [25], patchwork scheme [26], spatial quantizer scheme [27], etc., were defined in the original domain of the host signal, e.g. time for audio signals and space for digital images. Although these schemes have the advantage of being computationally less complex, they are generally less secure and robust [28].
- Frequency domain watermarking has the potential to offer better robustness and security since it is easier to analyze the host to determine importance ordering in this domain. Several popular transforms used for this purpose are the Discrete Cosine Transform (DCT) [22], the Discrete Fourier Transform (DFT) [29], the Discrete

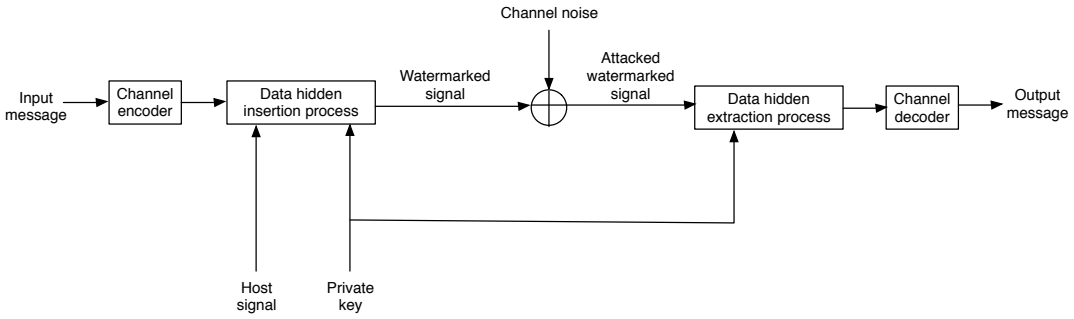

Fig. 1. Data hiding scheme mapped into a communications model

Wavelet Transform (DWT) [30], the Modulated Complex Lapped Transform (MCLT) [31], [32], etc.

As these transforms are similar to the ones used in other applications domains, it is possible to take advantage of predesigned standard processing modules, either in software or hardware. The selected transform is applied to the raw signal in spatial domain for the images or time domain for audio.

It is important to note that, regardless of their domain, most of the data hiding algorithms exploit the human perception characteristics. However, the Human Auditory System (HAS) and the Human Visual System (HVS) features are better modeled in the frequency domain. Therefore, the most of the data hiding systems in multimedia signals are developed in that domain.

D. Communication-Based Model of Data Hiding

Data hiding is, in essence, a form of communication. From this point of view, the message is transmitted using the host signal as a transmission channel [10]. In that model it is usual to carry out *channel coding* in order to keep the Bit Error Rate (BER) small enough [33]. Figure 1 shows a data hiding scheme mapped into a communications model.

The host signal can be assumed to be computed from the digital content to be marked in order to have a suitable set of features for the embedding. Therefore, the host signal can be a collection of coefficients of the digital content in a transformed domain, such as DCT, DWT or DFT. The other input to the insertion process is the private key, which is usually a secret shared with the decoder, and it allows the randomization of the embedding function in order to guarantee the security of the hidden information [8]. Once the marked content is generated, it undergoes the channel, where it is subject to attacks, which are usually modeled by means of a probabilistic channel. In a typical data hiding application the original host signal is not available to the data hidden extraction process. That scenery is known as data hidden blind extraction.

The communication model is used in emergent applications as cover communications. In this context, a example is as follows: the agent A took a picture using a smartphone, a secret message is hidden in the picture using a data hiding algorithm. Agent A email the marked picture to the agent B who recovery the secret message. If a third entity intercept the email, it will not see a message but only a picture. In this case, the picture was the communications channel.

III. Hardware Architectures for Data Hiding Schemes

In order to implement a real-time data hiding system two main platforms seem to be the natural election: Digital Signal Processors (DSP) and Field Programmable Gate Arrays (FPGA). Implementations on DSPs have been previously reported [4], [5]. Due to the programming strategies used on those implementations, the possible parallelism of the algorithms is not exploited. Technical outposts for DSP programming exist with the purpose of exploiting the parallelism of algorithms; nevertheless multi-channel processing in demanding tasks, such as video processing, is not straightforward. FPGA-based implementation of data hiding systems seems to be an interesting option since its capacity for parallel processing could allow multi-channel processing.

In the literature, hardware implementations of data hiding systems have been poorly reported. In [34] a data hiding system for speech bandwidth and its hardware implementation is proposed. The system uses data hiding techniques to transmit high frequency speech components in order to improve the speech quality in transmission systems. The hardware implementation is carried out using application software and one Fast Fourier Transform (FFT) implemented in a hardware platform. Due to the use of application software, the performance is limited in speed terms. A data hiding system using digital images as host signals and its hardware implementation is proposed in [35]. Performance results of the hardware implementation are superficially presented. However, the author claims that implementation using FPGA allows its application for real time multimedia data transmission. An FPGA implementation of a video watermarking algorithm and its comparison with a DSP implementation is reported in [36]. Implementation results for both FPGAs and DSP devices suggest that the FPGA is a better option in terms of processing speed, power consumption and device cost. In [37], hardware implementations of steganographic techniques that can be applied to documents, images and video are reported. According to the authors, implementation results show that

real time performance is guaranteed. In [38] an steganographic micro-architecture and its FPGA implementation is presented. The authors propose a video or audio steganographic model in which the hidden message can be composed and inserted in the cover medium in real time. Real time performance is demonstrated, with a reported throughput of 1.576 Mbps. In [39] a hardware architecture for an audio data hiding system is reported. Results show that the hardware architecture performs around 160 times faster that a software implementation.

A. Design Methodology

In order to develop a hardware architecture for a data hiding system several steps are typically followed:

- Firstly a theoretical analysis is carried out using MatLab as platform. In this step the precision requirements are found out. For example: in our experience, a Q15 precision is enough for audio applications in time domain, however, it is necessary validate through experiments when the domain is not the time and/or the signal has a different origin that audio.
- After of theoretical analysis it is a good practice realize a software implementation, typically in C language. These implementations are useful to experimentally validate the theoretical analysis of the algorithm and to obtain test vectors that will be used to validate the hardware architecture.
- Finally, the hardware architecture is designed. This process is more detailed in the next subsection.

B. Modules of a Typical Hardware Architecture for Data Hiding Applications

Figure 2 shows a generic block diagram of a hardware architecture that performs the data hiding process shown in figure 1.

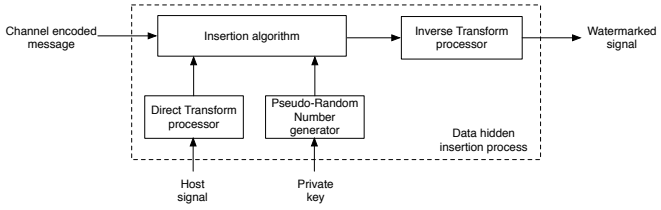

Fig. 2. Taxonomy of a hardware architecture for a typical data hidden insertion process

The Direct Transform and Inverse Transform processors compute the direct and inverse frequency coefficients of the host signal respectively. If the DFT computation is required the implementation could be carried out using a pre-designed core as [40]. However, sometimes it must be necessary to design the architecture from a fast algorithm as in [41].

As it is mentioned in the subsection II-B, a typical requirement is the security. Usually, in data hiding schemes security is provided by adding a pseudo-random number sequence to one or several participants in the insertion process, for example, to the host signal in the temporal domain [42] or in the frequency domain [31], [43]. In the figure 2 the Pseudo-Random Number Generator (PRNG) uses a private key as seed in order to provide key-based security to the data hiding scheme. Typically, PRNG is implemented with a Linear Feedback Shift Register (LFSR).

Finally, the architecture of the Insertion Algorithm module is always hand-designed. Most data hiding algorithms have low data dependency. However, others state-of-the-art algorithms as [44] have high data dependency due to it is necessary a set of previously watermarked samples in order to modify the actual sample. Insertion Algorithm module design almost always requires the largest effort in the workflow.

IV. Educational and Research Experience

In order to realize efficient hardware architectures for data hiding systems as described in Section III, an adequate background of the students is required. In this section we briefly describe the structure of INAOE's MSc Program on Computer Science as well as our experience looking for and adapting the skills of the students interested on developing hardware architectures for digital data hiding.

The MSc Program at INAOE consists of 10 curricular courses that are divided in three academic terms during the first year. Four courses during the first and second terms and 2 courses on a third short summer term. After taking these 10 courses, students start their thesis project on which they will work full time during the second year of the MSc Program.

As the MSc Program is intended to provide the students with a wide background on several areas of Computer Science, during the first term only one course is devoted purely to hardware related topic, computer architecture. During the second term, students can choose among several courses thus are able to train on the topics they need to complement their background knowledge on the topic they will work on for their thesis project. Those students that wish to work on hardware design related thesis take the course on Digital Design, plus some other courses related to the application domain, such as: Compression and Cryptography, Machine Learning, Image processing or Pattern Recognition. Finally, during the third term, they can broaden their knowledge of the application domain by taking advanced courses, including Digital Signal Processing and Data Coding.

At the MSc program on Computer Science there is a very strong emphasis on providing the students a complementary mix of background theory and application oriented material. This consistent emphasis provides a good environment for promoting research on applications from a systems-level viewpoint. In the next paragraphs we briefly describe the two main courses that students must take in order to develop their thesis work on architectures for Digital Data Hiding. The same mechanism is used for other application domains, it is only needed to change the application related courses.

- Computer Architecture. In this course, students are introduced to the basics of processor design with particular emphasis to architectures of Reduced Instruction Set Processors (RISC). The course covers from a review

of logic circuits to the construction of RISC processors and memory subsystems. The course also provides an overview of coprocessor design as well as Application Specific Instruction Processors (ASIP). At the end of this course, each student is asked to present an overview of a commercially available processor, such ARM, PowerPC or TMS320CXXXX processors. This helps the students to have a better understanding of set of features available in modern processor architectures.

- Digital Design. This course considers the design e implementation of FPGA-based hardware architectures. It covers the steps needed to implement architectures in today's FPGAs (design entry, functional simulation, logic synthesis, technology mapping, place and route and bit stream generation). Each student is assigned a project at the beginning of the course. Initially they are given a few weeks to understand the background of the application and the specific algorithm they will have to implement, parallel to this process the lectures are focused on providing a review of digital design techniques using the VHDL language. The tools used in the course are: ModelSim and Xilinx's System Generator and ISE. In addition students get access to a FPGA-based design board with Xilinx's devices (Virtex-II, Virtex-4, Virtex-5, Virtex-6, Spartan-3, and Spartan-6). The board assigned to each student depends on the requirements of their particular project.

In addition to the hardware related courses, the MSc Program includes a number of optative courses that allow the students to develop a strong background on a particular area of the computer sciences. For those students interested on doing their thesis project on hardware architectures for data hiding, some of the relevant courses are Data Compression and Cryptography, Image Processing, Digital Signal Processing and Data Coding. After the courses of the first year, students have developed the basics skills needed to begin their research work.

In general terms, our conclusions are similar to those expressed by the authors of other papers describing experiences of teaching hardware design to software engineers [45]–[49]. Based on our experiences from the last years, teaching the concepts of hardware design and reconfigurable computing to graduate students with a degree in computer science or computing engineering has proved to be a challenging task. In particular, students that have stronger background and experience developing complex software based systems find very difficult to grasp the low level concepts associated with hardware design. An additional hurdle is that in order to design a custom hardware-based system it is required to have a good understanding of the application domain, in the case of Digital Data Hiding, it is needed to have at least basic knowledge of concepts and algorithms of Digital Signal Processing (Audio, image, video, filtering, transforms, noise) and Information Theory (Entropy, coding, cryptography). Additionally good knowledge of numerical algorithms and methods is desired (Random numbers generators, polynomial equations solvers). An alternative to teaching software engineers the concepts of hardware design is to use high level design languages, such as System-C. However, after teaching digital design to two generation using System-C we found that software engineers ended up coding C like programs in System-C, which needless to say resulted in very poor architectures.

On the contrary, we have found that teaching all these concepts to students with a degree in electronics is usually less complicated, however they sometimes require to improve their programming skills, particularly in C or C++, because good programming skills are needed to implement embedded designs. Although the desirable background of the students that are developing hardware architectures for data hiding systems involves a degree in electronics, students with different degrees as in computer science or computing engineering can successfully develop hardware architectures too. However, based on our experience, it could be necessary a bigger effort from the students without a degree in electronics.

V. Conclusions

In this paper the field of digital data hiding and related hardware architectures implemented on reconfigurable hardware were introduced. The approach utilized to teach the skills needed to face thesis projects in that field at INAOE was also presented. Due to the main module in a custom hardware architecture of a data hiding system (the Insertion Algorithm module) is usually hand-designed in order to obtain the best possible performance, it is necessary that students have strong knowledge in digital design and background knowledge of the application. Based on our experience, the desirable background of the students developing hardware architectures for data hiding systems implies a degree in electronics. Unlike the experience at others Universities [50], teaching software engineers the concepts of hardware design using high level languages such as System-C has not been satisfactory. Therefore, it could be necessary a bigger effort from the students without a degree in electronics to develop efficient hardware architectures. On the other hand, good programming skills are required to implement embedded designs. In this case, students with a background in electronics sometimes require sharping their programming skills.

Acknowledgment

The authors would like to thank to Conacyt for financial support.

References

[1] F. Petitcolas, R. Anderson, and M. Kuhn, "Information hiding - a survey," in *Proceedings of the IEEE: Special Issue on Identification and Protection of Multimedia Content*, vol. 87, 1999, pp. 1062–1078.

[2] R. Tachibana, "Sonic watermarking," *EURASIP Journal on Applied Signal Processing*, pp. 1956–1954, 2004.

[3] T. Mizrahi, "Real-time implementation for digital watermarking in audio signals using perceptual masking," Signal and Image Processing Lab., Dept. of EE, Technion, Tech. Rep., 2002.

[4] J. J. Garcia-Hernandez, M. Nakano, and H. Perez, "Real time implementation of low complexity audio watermarking algorithm," in *Proceedings of the Third International Workshop on Random Fields and Processing in Inhomogeneous Media*, Guanajuato, Mexico, October 2005.

[5] ——, "Real time mclt audio watermarking and comparison of several whitening methods in receptor side," in *Proceedings of the Eighth IEEE International Symposium on Multimedia*, San Diego, Cal, USA, 2006, pp. 991–997.

[6] T. Furon, "A survey of watermarking security," in *International Workshop on Digital Watermarking*, Springer, Ed., vol. 3710 of Lecture Notes on Computer Science, 2005, pp. 201–215.

[7] D. Luo, N. Wu, C. Wang, Z. Lin, and C. Tsai, "A novel adaptive steganography based on local complexity and human vision sensitivity," *The Journal of Systems and Software*, vol. 83, pp. 1236–1248, 2010.

[8] M. Scagliola, "Digital watermarking methods applied to non-additive channels," Ph.D. dissertation, Politecnico di Bari, Italy, February 2010.

[9] S. Voloshynovskiy, S. Pereira, T. Pun, J. Eggers, and J. Su, "Attacks on digital watermarks: classification, estimation based attacks, and benchmarks," *IEEE Communications Magazine*, vol. 39, no. 8, pp. 118–126, August 2001.

[10] I. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital Watermarking and Steganography, 2nd Ed. (The Morgan Kaufmann Series in Multimedia Information and Systems)*, 2nd ed. Morgan Kaufmann, 2007.

[11] M. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia data-embedding and watermarking technologies," in *Proceedings of the IEEE*, IEEE, Ed., vol. 86, no. 6, June 1998, pp. 1064–1087.

[12] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*. CRC Press., 2004.

[13] I. Cox, M. Miller, and J.Bloom, *Digital Watermarking*. Morgan Kaufmann Publisher, 2003.

[14] G. Depovere, T. Kalker, J. Haitsma, M. Maes, L. de Strycker, P. Termont, J. Van-dewege, A. Langell, C. Alm, P. Norman, G. O'Reilly, B. Howes, H. Vaanholt, R. Hintzen, P. Donnelly, and A. Hudson, "The viva project: digital watermarking for broadcast monitoring," in *International Conference on Image Processing, 1999*, vol. 2, 1999, pp. 202–205.

[15] T. Kalker, G. Depovere, J. Haitsma, and M. J. Maes, "Video watermarking system for broadcast monitoring," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series,*, SPIE, Ed., vol. 3657, April 1999, pp. 103–112.

[16] P. Moulin and R. Koetter, "Data hiding codes," in *Proceedings of the IEEE*, IEEE, Ed., vol. 93, no. 12, 2005, pp. 2083–2126.

[17] M. Kutter and F. Hartug, *Introduction to watermarking techniques*, ser. Information hiding techniques for steganography and digital watermarking. Boston: Artech House., 2000.

[18] A. Hanjalic, G. Langelaar, P. van Roosmalen, J. Biemond, and R. Langendijk, *Image and video databases: Restauration, watermarking and retrieval*. Elsevier Academic Press, 2000.

[19] J. Fridrich, "Applications of data hiding in digital images," in *ISPACS '98 Conference*, 1998.

[20] P. Comesana, L. Perez-Freire, and F. Perez-Gonzalez, "Fundamentals of data hiding security and their application to spread-spectrum analysis," in *Information Hid- ing*, ser. Lecture Notes in Computer Science. Springer Verlag, 2005, vol. 3727, pp. 146–160.

[21] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, pp. 1673–1687, December 1997.

[22] I. Cox, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for images, audio and video," in *International Conference on Image Processing, 1996*, September 1996, pp. 16–19.

[23] C. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models," *IEEE Journal of Selected Areas in Communications*, vol. 16, pp. 525–539, May 1998.

[24] ——, "Perceptual watermarking of still images," in *IEEE First Workshop Multimedia Signal Processing*, IEEE, Ed., June 1997, pp. 363–368.

[25] E. Delp and R. Wolfgang, "A watermark for digital images," in *International Conference on Image Processing*, September 1996, pp. 219–222.

[26] J. Bruyndockx, J. Quisquater, and B. Macq, "Spatial method for copyright labeling of digital images," in *IEEE Workshop Image Processing*, September 1995, pp. 456–459.

[27] H. Lu, A. Kot, and J. Cheng, "Secure data hiding in binary document images for authentication," in *International Symp. Circuits and Systems*, May 2003, pp. 806–809.

[28] W. Zeng, H. Yu, and C. Lin, *Multimedia Security Technologies for Digital Rights Management*. Elsevier Academic Press, 2006.

[29] P. Moulin and A. Briassouli, "A stochastic qim algorithm for robust, undetectable image watermarking," in *International Conference on Image Processing*, October 2004, pp. 1173–1176.

[30] Q. Qin, W. Wang, S. Chen, D. Chen, and W. Fu, "Research of digital semi-fragile watermarking of remote sensing image based on wavelet analysis," in *IEEE International Symp. Geosciences and Remote Sensing*, September 2004, pp. 2542–2545.

[31] D. Kirovski and H. Malvar, "Spread spectrum watermarking of audio signals," *IEEE Trans. on Signal Processing*, vol. 51, no. 4, pp. 1020–1033, April 2003.

[32] J. J. Garcia-Hernandez, M. Nakano, and H. Perez, "Data hiding in audio signals using rational dither modulation," *IEICE Electron. Express*, vol. 5, no. 7, pp. 217–222, 2008.

[33] C. Fontaine and F. Galand, "How reed-solomon codes can improve steganographic schemes," *EURASIP Journal on Information Security*, vol. 2009, p. doi: 10.1155/2009/274845, 2009.

[34] F. Wu, S. Chen, and H. Leung, "Data hiding for speech bandwidth extension and its hardware implementation," in *2006 IEEE International Conference on Multimedia and Expo*, 2006, pp. 1277–1280.

[35] S. Mainty, A. Banerjee, and M. Kundu, "An image-in-image communication scheme and vlsi implementation using fpga," in *IEEE Indian annual conference (INDICON 2004)*, 2004, pp. 6–11.

[36] W. Irizarry-Cruz, "Fpga implementation of a video watermarking algorithm," Master's thesis, University of Puerto Rico, Mayaguez Campus, 2006.

[37] H. Y. Leung, L. M. Cheng, L. L. Cheng, and C. Chan, "Hardware realization of steganographic techniques," in *Third International Conference on International Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, vol. 1, 2007, pp. 279–282.

[38] M. Saeb and H. Farouk, "Design and implementation of a secret key steganographic micro-architecture employing fpga," in *The Conference on Design, Automation and Test in Europe*, I. C. Society, Ed., vol. 3, 2004.

[39] J. J. Garcia-Hernandez, C. Feregrino-Uribe, R. Cumplido, and C. Reta, "On the implementation of a hardware architecture for an audio data hiding system," *Accepted in Journal of Signal Processing Systems ISSN: 1939-8018 DOI: 10.1007/s11265-010-0503-8*, 2010.

[40] X. Inc., "Fast fourier transform v4.1," Xilinx. Inc., April 2007, http://www.xilinx.com/support/documentation/ip_documentation/xfft_ds260.pdf.

[41] J. J. Garcia-Hernandez, C. Feregrino-Uribe, and R. Cumplido, "Fpga implementation of a modulated complex lapped transform for watermarking systems," in *Proceedings of Reconfig 08, Cancun, Mexico.*, I. C. Society, Ed., 2008, pp. 367–372.

[42] B. Chen and G. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Information Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[43] J. J. Garcia-Hernandez, C. Feregrino-Uribe, R. Cumplido, and R. Parra-Michel, "Improving the security of fallahpour's audio watermarking scheme," *IEICE Electron. Express, ISSN 1349-2543*, vol. 7, no. 14, pp. 995–1001, July 2010.

[44] F. P. Gonzalez, C. Mosquera, M. Barni, and A. Abrardo, "Rational dither modulation: a high rate data-hiding method invariant to gain attacks," *IEEE Trans. on Signal Processing*, vol. 53, pp. 3960–3975, October 2005.

[45] R. Sass and D. Andrews, "Essential and elective topics: A proposal for the content of reconfigurable computing courses," in *The 1st International Workshop on Reconfigurable Computing Education*, Karlsruhe, Germany, 2006.

[46] M. Porrmann and J.-C. Niemann, "Teaching reconfigurable computing – theory and practice," in *The 1st International Workshop on Reconfigurable Computing Education*, Karlsruhe, Germany, 2006.

[47] C. Bobda, "Experiences in teaching reconfigurable computing at erlangen university," *New Trends and Technologies in Computer-Aided Learning for Ccomputer-Aided Design, IFIP International Federation for Information Processing*, vol. 192/2005, pp. 133–138, 2005.

[48] I. Skliarova, "A multimedia tool for teaching reconfigurable computing," in *Second International Conference on Computer and Electrical Engineering, 2009. ICCEE*, December 2009, pp. 204–208.

[49] L. Lagadec, "Building cad tools as an efficient learning for both ee and cs students," in *The 3rd International Workshop on Reconfigurable Computing Education, RC education*, Montpellier, France, April 2008.

[50] G. Wigley, "Teaching software engineers the basics of reconfigurable computing," in *The 1st International Workshop on Reconfigurable Computing Education*, Karlsruhe, Germany, March 2006.