Journal of ELECTRICAL ENGINEERING, VOL. 63, NO. 5, 2012, 289–295

BLOCK–WISE AUTHENTICATION METHOD FOR DIGITAL IMAGES

Tomaž Nahtigal — Andrej Žemva *

The ability to modify digital images can cause a serious problem in some applications. In this paper we propose a novel method derived from Wong's authentication scheme that is capable of authenticating JPEG images as well as uncompressed images, but is not a watermarking method. The method offers great flexibility in terms of the size of the authenticator and the time needed to generate the authenticator, thus making it suitable for real-time image processing. We demonstrate this by implementing it in a programmable FPGA circuit.

K e y w o r d s: authentication method, Holliman-Memon attack, global logo, tamper detection, block-wise authentication

1 INTRODUCTION

The rapid development of digital techniques, cost-effective digital storage devices and the widely spread personal computers and the Internet allow us to exchange and manipulate digital data with great ease. Analog data is in the process of being replaced by digital counterparts. The same applies to images. Digital images are easy to store, copy, edit and manipulate. They can be shared via computer networks, processed and stored in databases where they can be relatively simply managed.

Due to image editing software, ensuring authenticity of digital images poses a serious problem. The image editing software allows for malicious modifications of digital images which can be very difficult to detect. In some applications, authenticity of images is of vital importance. For instance medical images, images in news, images for evidence in court, etc. need to be protected in order to avoid false judgments.

Several methods, ranging from conventional cryptography to fragile watermarking, have been proposed to protect the authenticity of digital images. The methods differ in terms of the services they provide, that is tamper detection, localization and robustness against different image processing operations [1].

Despite the considerable number of proposed methods, only a few digital cameras equipped with authentication capabilities have emerged on the market. In [2–6], the authors propose a VLSI architecture for watermarking of digital images allowing for embedding of an invisible watermark in the image. The majority of papers proposing authentication methods are not concerned with the actual implementation of the authentication system. A workable authentication system that would be secure and ready for use in a wide range of applications is still to be made.

We believe that block-wise authentication methods are the most suitable for implementation because they process the image in a sequential manner. Wong [7] proposed a fragile block-wise watermarking scheme with tamper localization. The scheme processes 8×8 blocks in a sequential manner. As it requires no storing of the whole image, it is fast and efficient. Other schemes [8–12] derived from Wong improve its security, especially the resistance against the Holliman-Memmon attack.

Our goal is to design a method suitable for hardware implementation and enabling the system based on it to be secure, flexible and useful in real-life applications.

2 AUTHENTICATION SYSTEM

The task of the authentication method is to generate an authenticator (tag) from the data to be authenticated. The authenticator is a series of bits derived in a prearranged manner for the purpose of attesting the authenticity of an image. The authenticator can be stored in a separate file or file header.

The authentication method is implemented in the authentication system (Fig. 1). The system consists of an authentication unit and a verification unit. The authentication unit generates the authentication tag for the image and the verification unit verifies the authenticity of the image. The transition of the image from the digital camera to the end user can be seen as a sort of communication through an insecure communication channel.

The objective of the potential attacker however, is to produce a perfect forgery. To achieve this objective, the attacker needs to effectively mimic the workings of the authentication unit. In practice this is very difficult to achieve, so other types of attacks trying to fool the authentication system have been devised. Instead of creating a perfect forgery, their aim is to modify the original

* Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia, tomaz.nahtigal@fe.uni-lj.si, andrej.zemva@fe.uni-lj.si

DOI: 10.2478/v10187-012-0042-6, ISSN 1335-3632

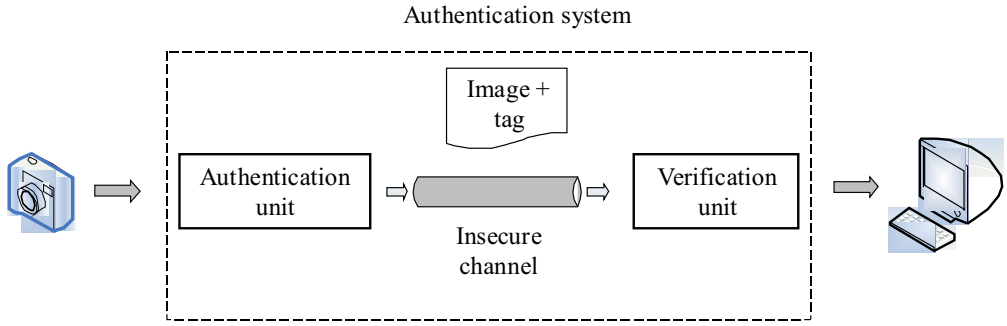


Fig. 1. Authentication system

image so that the authentication system recognizes it as being authentic.

2.1 Authentication unit

The first important requirement for an authentication system to be effective is that the authentication tag is generated as close as possible to the source of the image. This means that the authentication tag must be generated in the digital camera. If the authentication unit is not implemented in the digital camera, but is instead a program running on a computer, the attacker can modify the image before the authentication tag is generated and the changes in the image will remain undetected.

To implement the authentication unit in the digital camera, several design challenges need to be solved. The digital camera offers limited memory and limited processing capacity. The unit will also have to be capable of real-time image processing. An even greater challenge is imposed by the design of the authentication system that needs to be built into a digital camera of the mobile phone. Here factors like power consumption and cost play a much more important role. This means that special care should be taken in selecting the features of the authentication algorithm so as to allow for a balance between ease of implementation and the necessary level of security.

2.2 Verification unit

Another issue that strongly affects the complexity of the authentication system is accessibility of the verification unit. There are two possible implementations, i.e. an authentication system with a public access to the verification unit (public authentication system) and an authentication system with a private (restricted) access to the verification unit (private authentication system).

A public access to the verification unit means that anyone can acquire a verification unit from the device manufacturer. The attacker thus has access to the verification unit. It can verify the authenticity of any image and can generate any number of image-authenticator pairs. This is called the oracle attack. The strength of this attack depends on the output available to the attacker. The output can be either a binary yes/no for the whole image or it can be a bitmap with pixels or blocks indicated as authentic or tampered. Again, the attacker is interested in making undetectable changes.

For a method to be resistant to oracle attacks, it should incorporate a cryptographic primitive (encryption) so that the security of the method is assured with a secret cryptographic key and not just with the secrecy of the procedure itself. For the system to be practical, a verification unit should be capable of verifying an image from any camera of the same manufacturer. This implies that the cameras and the verification units should share a limited number of pre-chosen secret keys or that a more elaborate key management scheme should be adopted. The problem of this approach is in the responsibility for the management of the secret keys which is with the manufacturer.

The private authentication system is more secure; it prevents the oracle attack and information leakage. Though the authentication method itself can therefore be simpler and easier to implement, the verification procedure is more complicated. In order to supervise verification, a trustworthy authentication (verification) center must be established.

For our method to cover the widest possible range of applications, we chose to design it for the worst case scenario, meaning a method suitable for a public authentication system.

2.3 Block-wise authentication methods

A practical authentication method needs to incorporate a cryptographic primitive and also be fast and efficient, so it can be implemented in a digital camera. We believe that block-wise authentication methods are the most suitable candidates, because they process the image in a sequential manner. The pixels are read from the image sensor sequentially, so it is convenient to process the image in a sequential manner. To process the image as a whole, we would have to first read the image from the image sensor and then store it in memory. This requires extra time and memory.

One of the first fragile block-wise watermarking schemes with tamper localization was proposed by Wong [7]. In this scheme, an image is divided into non-overlapping blocks and watermarking is performed for each block independently. The seven most significant bits (MSBs) of all pixels in a block are hashed using a secure key-dependent hash. The hash is then XORed with a chosen binary logo

(a) (b) (c)

Fig. 2. Authentication system

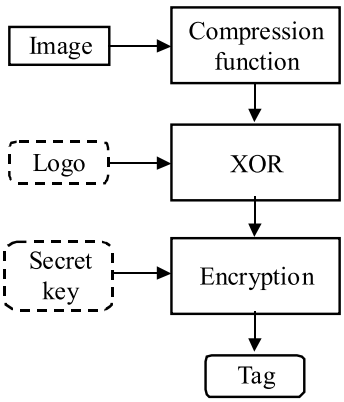


Fig. 3. General authentication scheme

and inserted into the LSBs of the same block. The verification process starts in the reverse order by calculating the key-dependent hash of the seven MSBs in each block and XOR operation is performed with the LSBs. The tampered blocks can be found by comparing the output with the used logo.

Block-wise authentication schemes with non-overlapping independent blocks are vulnerable to a certain kind of attack. If a set of images is authenticated with the same key, it is possible to modify an arbitrary image to be authentic. The attacker divides the image into non-overlapping blocks and for each of them performs a search in the set of authentic blocks. The original block is replaced with the most similar one to maintain the perceptual quality of the forged image. This attack is known as the Holliman-Memon attack or collage attack. There have been various countermeasures proposed in the literature [8–12] to resist the Holliman-Memon attack. In general, the countermeasures try to eliminate block independence so that block swapping is no longer possible.

2.4 Localization

Localization is one of the features a lot of researches have been focused on. The argument for localization is that the knowledge of when and where the data has been altered can be used to infer the motive for tampering and identifying the culprits responsible.

In the case of block-wise authentication schemes, we have to be very careful what kind of conclusions we draw from this information. The only thing we can be certain of is that the non- tampered parts of the image are authentic. For instance, the attacker changes a part of the image so that the verification unit fails to recognize it as authentic but the visual information of the image remains the same. The conclusions we draw from the verification results are therefore misleading. We see what the attacker wants us to see. An example is shown in Fig. 2, where Fig. 2a is the original image, Fig. 2b shows the tampered images and Fig. 2c the verification results. The plate number in Fig. 2a is the same as in the lower image of Fig. 2b, but the verification result shows that the image has been tampered with. Both images in Fig. 2b have been tampered with, but the visual information of

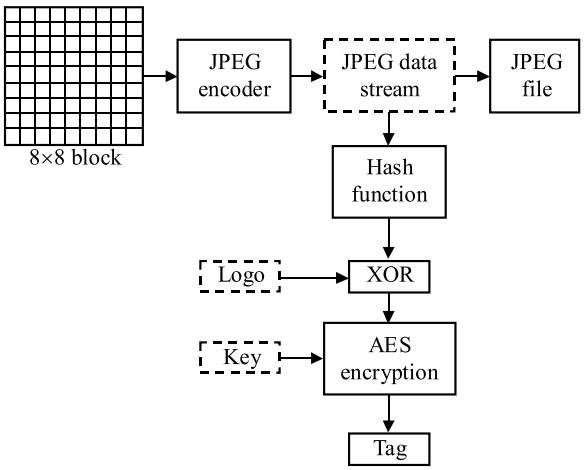


Fig. 4. General authentication scheme

the lower image remained the same. Based on the verification result, we cannot infer which plate number in Fig. 2b has been changed.

In order to make useful conclusions based on localization of the tampered regions, the attacker must be unaware that the changes it makes will be detected. In practice, if the authentication system is available on the market, this is impossible to achieve. This means that there is no real use for localization in block-wise authentication schemes.

3 PROPOSED METHOD

In this section we propose a novel authentication method based on Wong's localization scheme [7]. We extracted the general features from Wong's scheme and used them to construct a general authentication scheme presented in Fig. 3. The authentication scheme is in general a MAC (message authentication code) algorithm based on universal hash functions. Such an algorithm consists of two building blocks, an efficient keyed compression function that reduces long inputs to a fixed length and an encryption function that encrypts the hash.

To construct the authentication scheme we use a family of (hash) functions
$H = \{h\colon \{0,1\}^{\alpha} \to \{0,1\}^{\gamma}\}$ a family of secure pseudo-random functions
$F = \{f\colon \{0,1\}^{\tau} \to \{0,1\}^{\tau}\}$ and a logo $L = \{0,1\}^{\mu}$.
The scheme $\Sigma = (K, MAC)$ is defined as follows:

Function $K(\,)$:

1: Select function f from the family of functions F, $f \leftarrow F$,
2: Select function h from the family of functions H $h \leftarrow H$, 3:
3: Return (f, h).

Function $MAC_{(f,h)}(I, L)$:

1: Break image I into m-bit blocks $l = l[1] \ldots l[n]$,
2: For $i = 1 \ldots n$ do
Compute, $y_i = f\big(l[i] \oplus h(I[i])\big)$,
3: $Tag = y_1 \| y_2 \| \ldots y_n$,
4: Return (Tag).

The key-space for this authentication scheme is $Key = H \times F$. A random key for the MAC is a random hash function $h \in H$ together with a random function $f \in F$.

In our scheme, the input data is first compressed and then XORed with the logo. The result is further encrypted with a secret key and stored in the file header or in a separate file. To verify the authenticity of the image, the verification unit compresses the received image and XORs it with the decrypted tag. The verification unit decides if the image is authentic based on the resulting logo.

The logo proposed by Wong was either a binary image with a graphical meaning or a randomly generated black and white pattern. The idea of the logo was extended by Friedrich *et al* [12]. They proposed a symmetry structure, allowing the logo to carry additional information, such as block index, image index, author ID, *etc.* We propose a similar approach with an improved security presented in the following section. Our logo also carries information about block index, image index, camera id, time and date the image has been taken, camera settings (aperture, shutter speed, focal length, etc.) and possibly GPS coordinates. The additional information can help the end user determine the circumstances under which the image has been taken and make it easier to interpret the visual information of the image.

In this general scheme, the format of the input and realization of the compression function are left undetermined. The output of the scheme is a part of the authenticator or the complete authenticator. The input can vary from a single pixel to the whole image. There are several ways in which we can apply the general scheme regarding the input and output, each allowing for a different size of the authenticator and the speed of the method.

3.1 Block-wise authentication method

One of the possible implementations of the general scheme is presented in Fig. 4. The encryption unit uses the 128 bit AES algorithm to encrypt the data. AES has a fixed block size of 128 bits and uses a 128 bit secret key. The reason we use a block-wise authentication method is not localization but speed. Because the pixels are read from the image sensor sequentially, it is convenient to process the image in a sequential manner. To process the image as a whole, we would have to first read the image from the image sensor and then store it in memory. This would require extra time and memory.

3.2 Compression function

The compression function consists of two steps. In the first step the block is compressed using the JPEG algorithm. The output is then passed on to a universal hash function for further processing. The role of the hash

(a) (b) (c)

Fig. 5. (a) Original image, (b) tampered image, (c) results of verification

function is to take the variable-sized input from the JPEG compression and return a fixed-sized output suitable for the next step in the authentication algorithm.

By using the JPEG algorithm in the compression function we actually authenticate the JPEG representation of the image. The end result is an image-authenticator pair with the image in the jpg format. Our method is therefore tolerant to JPEG compression with a single quality setting. Because JPEG compression is lossy, a subsequent recompression of the image is not tolerated as it would cause the verification unit to recognize the image as unauthentic. If we wanted to authenticate an uncompressed image, we would only have to omit the first step in the compression function. The end result would be an image-authenticator pair with the image in raw format.

3.3 Global logo

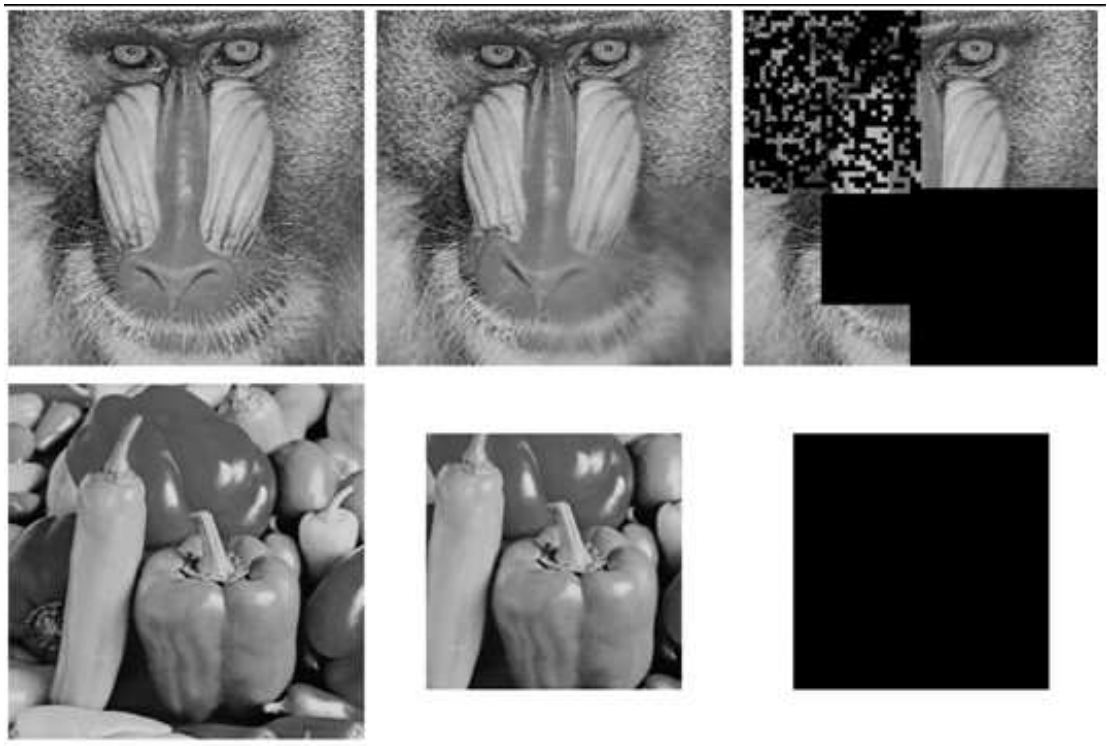
Our countermeasure against the Holliman-Memon attack is a variation of the content origin authentication [12]. The logo carries the local block information as well as parts of information which, when put together, form a global pattern throughout the image. The logo consists of three bit fields: block index, image index and content data. The block index is a unique predefined number assigned to the block. The image index is a random number generated at creation of the image and is also unique to the image. The verification unit does not know the image index, but knows that the image index is the same in all the blocks. The content data field is a fragment of the data string consisting of the camera id, time and date the image has been taken, camera settings (aperture, shutter speed, focal length, etc.) and possibly GPS coordinates. The data string is larger than the content data field, so in order to reconstruct the data string, we must join together several consecutive content data fields. The string is repeated throughout the image. The block and the image are seen as being authentic if the block index corresponds to the position of the block, the image index is the same in all the blocks and the same data string is repeated throughout the image.

4 RESULTS AND DISCUSSION

In this section we discuss security of our method and present simulation results for the Holliman-Memon attack. We also provide synthesis results for the authentication unit.

4.1 Simulation results

Our method was implemented and verified in Matlab. The verification of the method was performed on a standard test image set [13, 14]. The images from the set underwent different image manipulations (block interchanging, pixel manipulation, noise addition, low-pass filtering, cropping ...). Two of the test images are shown in Fig. 5a and the tampered ones in Fig. 5b. In the latter, we corrupted some of the pixels and interchanged some of the blocks in the images. The results of the verification are shown in Fig. 5c with the unauthentic blocks marked



(a) (b) (c)

Fig. 6. (a) Original image, (b) tampered image, (c) results of verification

in black. As seen, the verification unit detects the tampered areas and the collage attack is not possible as the blocks are no longer independent.

Figure 6 shows test images that underwent other image manipulations along with results of the verification. The pepper image in Fig. 6 was cropped. The result of the verification is a black image. The mandrill image was divided into quadrants. A low-pass filter was used in the lower right quadrant. The lower left quadrant was distorted. In the upper left quadrant noise was added and the upper right quadrant was left unchanged (Fig. 6b). The tampered and non tampered areas were detected correctly in all the test images.

4.2 Implementation

The proposed method can be integrated in the JPEG compression unit already present in the digital camera, therefore consuming fewer resources. The main part of computation is required for AES encryption. The method generates an authenticator for every block. By adequately adjusting the hash function, it can compress more than one block. This enables adaptation of the size of the authenticator and the speed of the method. The only thing that is lost is the resolution of the localization.

To validate our approach in terms of the speed and size of the authentication unit, we implemented the authentication unit in a programmable FPGA circuit. The unit is comprised of the hash function, JPEG compression unit and AES encryption. The AES encryption core constitutes the main part of the authentication unit regarding the area and computation intensity. The unit was modeled using VHDL, the functional simulation was carried out using Modelsim XE III 6.3c and the design was synthesized using Xilinx ISE 11.1. The target device was xc3s200 from the Spartan3 family. The device utilization summary is presented in Tab. 1.

Table 1. Device utilization summary

Device	Size/Area	Speed/Performance
XS3C200	1604 Slices (81%)	60MHz($\sim$ 400 Mbit/sec)

To have the verification unit integrated into the digital camera, our primary design goals were a small area and low power. The resulting design provides an estimate on the requirements for a workable authentication unit. Especially the design of the AES core allows for a variety of possible implementations. There have been several hardware implementations of AES [15, 16] proposed, yielding good performances. Ultimately, the best architectural decision is to select the design of the smallest possible area meeting the throughput requirement for the whole system. Following the above, we can assume that the authentication unit is suitable for hardware implementation.

5 CONCLUSION AND FUTURE WORK

In this paper we presented a novel authentication method derived from Wong's authentication scheme that is capable of authenticating JPEG images as well as uncompressed images. The tag consists of a number of block codes carefully chosen to protect regions of interest in the image and to enable various levels of accuracy. Our logo also carries information about the camera id, time and date the image has been taken, camera settings (aperture, shutter speed, focal length, etc.) and possibly GPS coordinates. The additional information can help the end user determine the circumstances under which the image has been taken and make it easier to interpret its visual information. We propose an efficient authentication method with adjustable security that can be combined with the JPEG compression algorithm. The system being specifically designed to tolerate JPEG compression does not tolerate other types of acceptable modifications. By skipping the first step in the compression function, the system is also capable of authenticating uncompressed images. Compared to Wong's scheme, our scheme outperforms it with improved security and flexibility.

References

[1] HAOUZIA, A.—NOUMEIR, R.: Methods for Image Authentication: a Survey, Multimedia Tools and Applications, 2007.

[2] ADAMO,—MOHANTY, S. P.—KOUGIANOS,—VARANASI, V.: VLSI Architecture for Encryption and Watermarking Units Towards the Making of a Secure Digital Camera, in Proc. of the IEEE International SOC Conference, 2006, pp. 141–144.

[3] BLYTH, P.—FRIDRICH, J.: Secure Digital Camera, in Proc. Digital Forensic Research Workshop, 2004.

[4] NELSON, G. R.—JULLIEN, G. A.—PECHT, O. Y.: CMOS Image Sensor with Watermarking Capabilities, in Proc. of the IEEE International Symposium on Circuits and Systems, 2005, pp. 5326–5329.

[5] MOHANTY, S. P.—KOUGIANOS, E.—RANGANATHAN, N.: VLSI Architecture and Chip for Combined Invisible Robust and Fragile Watermarking, IET computers and digital Techniques **1** No. 5 (2007), 600-611.

[6] KARTHIGAI KUMAR, P.—BASKARAN, K.: An ASIC Implementation of a Low Power Robust Invisible Watermarking Processor, Journal of Systems Architecture **57** No. 4 (2011), 404–411.

[7] WONG, W.: A Public Key Watermark for Image Verification and Authentication, in Proc. of the IEEE International Conference on Image Processing, vol. 1, 1998, pp. 455–459.

[8] PUHAN, N. B.—HO, A. T. S.: Secure Authentication Watermarking for Localization Against Holliman-Memon Attack, Multimedia Systems **12** No. 6 (2007), 521–532.

[9] WONG, P. W.—MEMON, N.: Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification, IEEE Trans. Image Process. **10** No. 10 (2001), 1593–1601.

[10] CELIK, M. U.—SHARMA, G.—SABER, E.—TEKALP, A. M.: Hierarchical Watermarking for Secure Image Authentication with Localization, IEEE Trans. Image Process. **11** No. 6 (2002), 585–595.

[11] HOLLIMAN, M.—MEMON, N.: Counterfeiting Attacks on Oblivious Block-Wise Independent Invisible Watermarking Schemes, IEEE Trans. Image Process. **9** No. 3 (2000), 432–441.

[12] FRIDRICH, J.: Security of Fragile Authentication Watermarks with Localization, in Proc. SPIE, vol. 4675, 2002, pp. 349–356.

[13] SCHAEFER, G.—STICH, M.: UCID – An Uncompressed Colour Image Database, Technical Report, School of Computing and Mathematics, Nottingham Trent University, U.K, 2003.

[14] http://www.imageprocessingplace.com/root_files_V3/image_databases.htm.

[15] KOSARAJU, N. M.—VARANASI, M.—MOHANTY, S. P.: A High Performance VLSI Architecture for Advanced Encryption Standard(AES) Algorithm, in Proc. of the 19th IEEE International Conference on VLSI Design, 2006, pp. 481–484.

[16] GOOD, T.—BENAISSA, M.: AES on FPGA from the Fastest to the Smallest, in Proc. of CHES, 2005, pp. 427–440.

Received 13 October 2011

Tomaž Nahtigal is a PhD student at the Faculty of Electrical Engineering, University of Ljubljana, Slovenia. He received his BSc in Electrical Engineering at the same university in 2007. His research interests include design and verification of digital systems, development of video and imaging applications and HW/SW co-design.

Andrej Žemva received his BSc, MSc and PhD degrees in electrical engineering from the University of Ljubljana in 1989, 1993 and 1996, respectively. He is Professor at the Faculty of Electrical Engineering. His current research interests include digital signal processing, HW/SW co-design, ECG signal analysis, logic synthesis and optimization, test pattern generation and fault modeling.