Hardware realization of a lightweight 2D cellular automata-based cipher for image encryption

Cesar Torres-Huitzil
Information Technology Laboratory, CINVESTAV-Tamaulipas
Parque Científico y Tecnológico TECNOTAM, Ciudad Victoria, Tamaulipas, México
Email: ctorres@tamps.cinvestav.mx

***Abstract*—Cryptography is one of the fundamental techniques used to secure data storage and transmission of sensitive multimedia data exchanged between different communicating parties. Several mobile and embedded devices are equipped with high resolution digital cameras and resources for multimedia management but most of them lack mechanisms to protect image content due to the additional computational cost of cryptographic algorithms and the inherent power constraints in such resource-limited devices. In this paper, a two-dimensional (2D) cellular automata (CA) based stream cipher for color image encryption is presented. A CA is used to generate a good quality pseudorandom bit sequence that is used in a stream cipher scheme for color image encryption that outperforms related works. The proposed scheme, prototyped into a field-programmable gate array (FPGA) device, allows image encryption with reduced hardware resource utilization suitable to be embedded as a low-power component in mobile computing systems.**

I. Introduction

Due to the wide spread use of embedded, mobile and pervasive computing devices aiming at providing information and services everywhere-everytime, practical methods, such as cryptography, watermarking and steganography, are becoming of primordial interest so as to provide security and authenticity of sensitive multimedia data exchanged between different communicating parties [1][2]. Embedded and mobile devices such as smartphones are equipped with high resolution digital cameras and resources for multimedia management. Users might capture images and share them with others by sending them via any wireless communication link. Cryptographic techniques can be used in such devices to provide security to image content by making the unauthorized recovery of the information a difficult, if not an impossible task.

Several encryption techniques and schemes have been proposed in the literature for multimedia applications targeting different domains and performance criteria [1][3]. Most works have addressed image encryption with cryptographic ciphers in desktop applications, but high-resolution images and video rates in embedded and mobile computing systems still require lightweight schemes and/or a considerable computational power that is not available. Thus, compact specialized hardware architectures are desirable in order to circumvent, at least partially, this problem. In this context, stream ciphers are suitable for embedded applications due to their moderate computational complexity. Such ciphers require of good statistical pseudorandom bit streams but most importantly, they rely on the unpredictability of the bits in the sequence.

978-1-4673-4900-0/13/$31.00

A variety of chaos-based image encryption schemes has been proposed based on the unpredictable and random nature of chaotic systems [4][5][6], but they are not well suited for hardware implementation due to high numerical precision requirements. Stream ciphers targeted to hardware commonly use linear feedback shift registers (LFSRs) as pseudorandom number generators (PRNGs) due to simple bitwise arithmetic-logic operators organized in one dimensional structures. However, LFSRs must be combined with non-linear processing to improve their statistical properties adding an extra implementation cost [7][8]. On the other hand, CA have shown better randomness quality than similar complexity LFSR structures [9]. Yet CA are simple, regular, locally interconnected, modular structures, and potentially highly parallel and scalable with little hardware cost.

In this work a compact image encryption hardware implementation based on pseudorandom bit sequences generated by 2D CA is presented, following a stream cipher scheme, targeted to embedded and mobile applications. Implementation results show a good compromise between performance and resource utilization that outperforms previous related works. The rest of the paper is organized as follows. Section II provides the fundamentals of CA to generate pseudorandom bit streams and the general stream cipher scheme for color images based on CA. Section III describes some details of the hardware implementation. Section IV presents experimental results targeted to FPGA technology and a discussion on the quality of the encrypted image in relation to the random sequences generated by the 2D CA.

II. Cellular automata and image encryption

A. Cellular automata and pseudorandom streams

Cellular automata are dynamical systems in which both space and time are discrete. A cellular automaton consists of neighboring cells arranged in regular arrays or lattices whose behavior evolves in discrete time steps. The connectivity among cells is defined locally and a transition function governs the cell behavior. Figure 1 shows a conceptual view of a 2D CA. The number of cells used in the grid can be adjusted depending on the expected outcomes of random numbers. Each cell holds a state, which corresponds to a particular bit in the produced random number or stream. After all cells update their states, a new random number or stream is obtained by collecting the bits scattered among the cells.

Two CA configurations reported in the literature were used to evaluate the quality-performance trade-off for image encryption implementation. An homogenous non-local connections

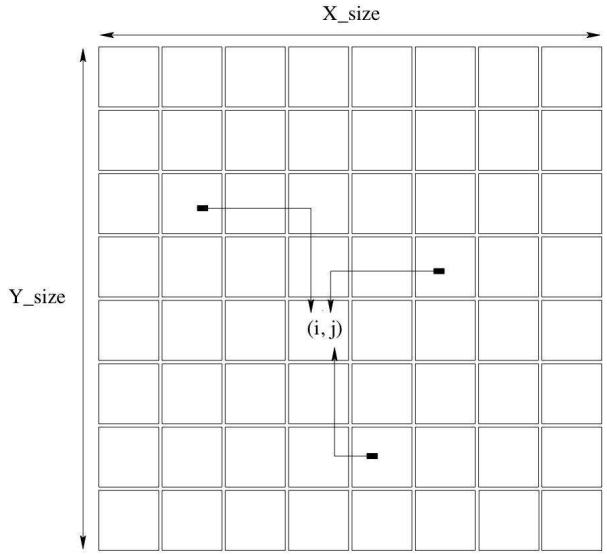


Fig. 1: Conceptual view of 2D CA-based PRNG.

CA proposed in [10], based on a grid of 8×8 with periodic boundary conditions, presents good randomness quality. The connectivity rule is as follows $\{2n2w, c, n2e, 2se\}$ as shown in figure 1. Authors used the compass directions n (north), s (south), e (east), and w (west) to indicate unit displacement along columns and rows relative to a center cell c. The 2D CA proposed in [11], is another PRNG that produces high quality sequences of random numbers. The rule numbering for the inhomogeneous CA with null boundaries is as follows. Let s be the state of the cell at position (i, j), at time t. Its state at the next time step, is then computed as:

$$s_{i,j}(t+1) = x \oplus (c \cdot s_{i,j}(t)) \oplus (n \cdot s_{i-1,j}(t)) \oplus (w \cdot s_{i,j-1}(t)) \oplus (S \cdot s_{i+1,j}(t)) \oplus (E \cdot s_{i,j+1}(t)) \quad (1)$$

where $\oplus$ and $\cdot$ are the xor and and logical operators, respectively, and x, c, n, s, w, and e are binary variables indicating whether the respective neighboring cell state is taken into account or not. The rule numbering of a cell is then given by the 6-bit string $xcnwse$. Four different transition rules are enough to obtain good randomness according to the results generated by an evolutionary approach [11].

Since stochastic heuristic search techniques were used to design the cited PRNGs [10][11], there is no guarantee that the results are optimal and hence better 2D CAs cannot be contructing by modifying some parameters. In a previous work [12], we evaluate the nature of 4-neighbor neighborship as a mean to improve the randomess quality, see section IV.

B. CA-based image encryption/decryption scheme

A stream cipher breaks the message M into a stream of succesive bits (bytes) $p_1, p_2, p_3, ...$ and enciphers each bit with a random stream of bits (bytes) $k_1, k_2, k_3, ...$ generated by a key stream generator such that:

$$E_K(M) = E_{k_1}(p_1)E_{k_2}(p_2)E_{k_3}(p_3)... \quad (2)$$

More precisely, for image encryption, assume the original image I is in uncompressed format and each RGB value of a pixel, within the range $[0, 255]$, is represented by 8 bits. Let P a pixel of I, and E_K an enciphering algorithm. The main transformation to obtain a cipherpixel C is:

$$C = E_K(P) \quad (3)$$

where K is the key of the transformation which distinguishes a particular encryption in a family of transformations using the same enciphering algorithm [2].

To recover the original pixel, a deciphering function D, using the same key K, is defined as the inverse of E:

$$P = D_K(C) = D_K(E_K(P)) = E_K^{-1}(E_K(P)) \quad (4)$$

A common enciphering operation, and the one used in this work, is the *exclusive or* operation $\oplus$, reducing equation 3 to:

$$c_i = k_i \oplus p_i \quad (5)$$

where c_i is the *ith* pixel of the cipher-image and the original pixel p_i is obtained as follows:

$$p_i = c_i \oplus k_i = (k_i \oplus p_i) \oplus k_i \quad (6)$$

III. FPGA-BASED IMPLEMENTATION

A general block diagram of the proposed hardware implementation is shown in figure 2. The architecture follows a symmetric private key stream cipher approach to encrypt/decrypt images, i.e., the same key is required for encryption and decryption. The image pixels are read from the input memory and then sent to a bitwise combinational function that use a random stream to encrypt pixels. A bitwise exclusive or is used as the combinational function for encryption, however, in the architecture any combinational function can be used and defined at synthesis step. For the current implementation 24 2-input xor functions are used, one per bit in each image color plane. The input and output memory addresses are generated by an address generator unit which is coordinated with other components by a control module to properly generate the control and synchronization signals.

The core component of the architecture is the 2D CA random stream generator and the following parameters of its hardware description language (HDL) model can be configured: the size of the grid, the transition function of each cell, the neighborhood, and boundary conditions. The initial state of the CA is used as the key for encryption. Currently, this component includes two CA models for random stream generators as described in section 2, but any CA can be used without affecting performance or FPGA resource utilization. The storage elements and the computation granularity of FPGA resources are well suited to support an efficient implementation of the basic cell and the regular structure of CAs. The transition function is mapped into a single FPGA Look-up-table (LUT). Since most up-to-date FPGAs contain 4 to 6 input LUTs, transition functions of 4 to 6-neighbor can be implemented efficiently. Yet, the cost of a transition function does not depend on the complexity of the function itself and does not impose any significant delay on the enciphering architecture. Recall that FPGA is used as a prototyping device, thus the cipher can also be implemented as an specific circuit.

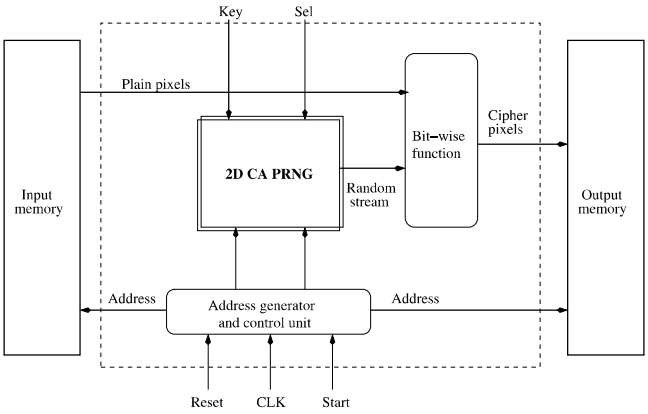
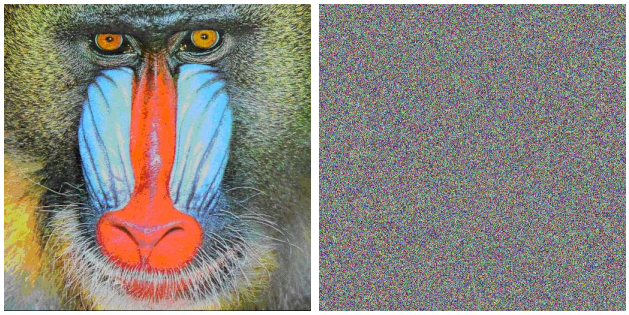


Fig. 2: Block diagram of the CA-based stream cipher architecture for RGB color images.



(a) (b)

Fig. 3: RGB image encryption and decryption experimental result: (a) plain-image, (b) cipher-image.

IV. Experimental results and evaluation

The proposed architecture has been tested with various 512×512 RGB images. Figure 3 shows the Baboon image and the corresponding cipher image. The parameters taken for experimentation are: a 64-bit key is used as the initial state for the 8×8 CA with the architectural details presented in [10], and the xor function as the enciphering operation. It is clear from figure 3 that the cipher image is visually indistinguishable from the original image since it presents completely disordered pixel values all over the input image plane. Thus, the statistical characteristics of plain-images are enhanced by the 2D CA in such a manner that the generated cipher-images have fairly uniform RGB distributions.

The histogram analysis on the test images, not shown for space considerations, shows that the cipher image histogram for the RGB planes is fairly uniform and significantly different from the original image histogram. Another statistical test for cipher images is correlation of adjacent pixels. In ordinary images, each pixel is usually highly correlated, with its adjacent pixels either in horizontal, vertical or diagonal directions [13]. Table I shows the correlation coefficients of the input and cipher image shown in figure 3. Different images have been tested, and similar results are obtained. The cipher image adjacent pixels retain small correlation coefficients in all directions and they are competitive or even better than chaos-based encryption techniques [4][5][13]. Moreover, for similar approaches the correlation coefficients are in the order of hundredths and the required resources for the different-sized CA-based PRNGs used range from 95 to 773 of the total number of slices available in the Virtex XC2V1000 device [14]. It can be noticed that the proposed method requires less hardware resources than its counterpart, see table II, thanks to the improved statistical properties of the CA.

TABLE I: Correlation coefficients of adjacent pixels in the original and the cipher image.

Correlation coefficient	Original image	Cipher image
Horizontal	0.9821	0.0018
Vertical	0.9773	0.0038
Diagonal	0.9680	0.0023

A 2D CA configuration that keeps most of the parameters proposed in [10] was used as the PRNG, except that different asymetric 4-neighbor neighborhoods were explored. The 4 neighbors were restricted within a 5×5 area around a center cell to reduce the searching space. After an exhaustive search, the neighborhood that provides the best results, according to standard statistical tests, is given by $\{2sw, 2s2w, ne, 2e\}$, see [12] for details. The PRNG performance was measured by applying the Diehard suite, consisting of 18 different and independently statistical tests. Most of the tests return a $p-value$, which should be uniformly distributed if the input stream contains truly independent random bits. A $p-value$ was classified as good if $0.025 < p < 0.975$ and rejected otherwise. Data for each Diehard run were collected from the 2D CA during 3 million cycles, giving a total of 3 million 64-bit words [10][12].

Figure 4 shows the Diehard test result digest with the lower and upper bound indicating the range of acceptable $p-values$, region between the vertical lines. Figure 5 shows that the $p-values$ obtained from the test suite roughly approximate to a uniform distribution improving the results presented in [10]. The CA PRNG is configurable and other configurations can be used to achieve better statistical properties if desired. For instance in might be interesting to increase the size of the CA in order to have larger keys (128 bits) in order to avoid brute-force attacks. Recall, that the proposed CA PRNG uses a 4-neighbor transition function but it can be up to 6-neighbor in the Spartan6 FPGA technology at the same hardware cost.

The proposed architecture was modeled in VHDL and synthesized, placed, and routed using the Xilinx ISE 13.1 tool suite targeted to a Spartan6 device. It was fully validated in the Digilent Atlys prototyping board with a stereo camera module that features two Aptina MT9D112 2-megapixel CMOS digital image sensors. The FPGA resource utilization, shown in table II, makes the architecture potentially useful in low-power mobile embedded applications and suitable to work for video stream systems in very high speed local networks. The maximum clock frequency is 468.60 MHz (2.134 ns period). Since the architecture produces a 24-bit RGB cipher pixel per clock cycle, the throughput of the cipher is 11.248 Gbps. Thus, the time required to process a 512×512 input RGB image is around 560 microseconds which is appropriated even for real-time full HD-1080p video [15].

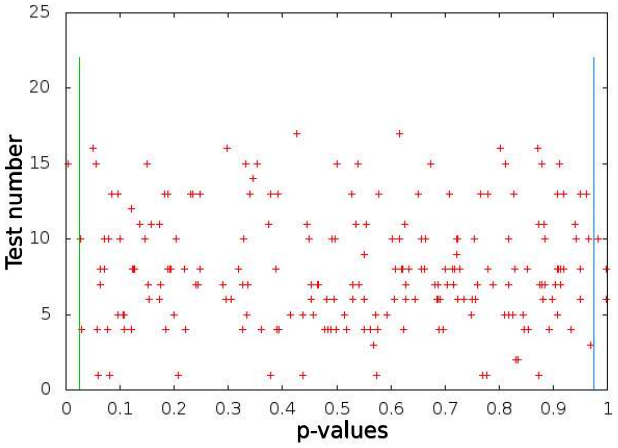


Fig. 4: p-value distribution of the CA-based PRNG.

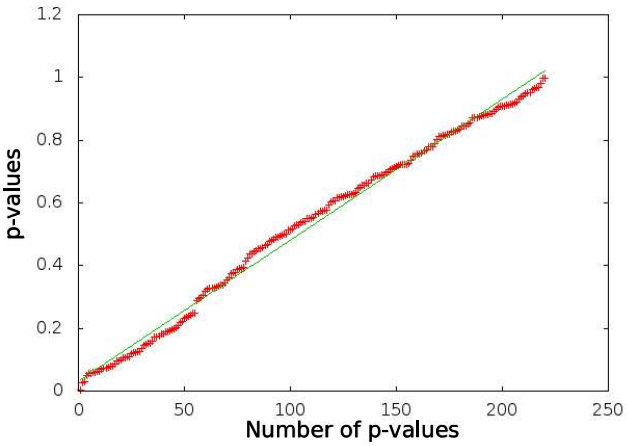


Fig. 5: p-value regression of the CA-based PRNG.

V. Conclusions

This paper has presented a compact CA-based hardware implementation for RGB image encryption/decryption. Experimental tests and analysis, demonstrate that the architecture is able to provide security to the image content suitable for embedded applications and that outperforms similar approaches. The CA-based stream scheme is able to resist standard statistical and key analysis attacks. Since the user key is in 64 bits, the key space is rather sufficient large to resist the brute-force attack with the current computer technology. Moreover, the key space can be easily expanded in order to satisfy the future requirement without significative implementation costs. The proposed encryption scheme is also key-sensitive, meaning that a tiny change in the key will cause a significant change in the output. The FPGA implementation shows that the proposed scheme is suitable to be embedded as a low cost coprocessor for low-power mobile applications, in distributed scenarios with low-power sensor devices. where the encryption and decryption time should be short. For example the proposed cipher can be used to secure fingerprint images, yet as a technology for embedded biometric encryption systems.

TABLE II: FPGA hardware resource utilization using a Xilinx Spartan xc6xlx45t-4fgg484 as a target device

Module	# Slices	# LUTs	# Flip-flops
2D CA PRNG	17	64	64
Bit-wise function	17	24	0
Control unit	8	32	32
Complete architecture	35	120	96

Acknowledgment

This work was supported by CONACyT, Mexico, under research grant No. 99912.

References

[1] C.-P. Wu and C.-C. Kuo, "Design of integrated multimedia compression and encryption systems," *Multimedia, IEEE Transactions on*, vol. 7, no. 5, pp. 828–839, oct. 2005.

[2] M. Tomassini, "Cryptography with cellular automata," *Applied Soft Computing*, vol. 1, no. 2, pp. 151–160, Aug. 2001.

[3] M. Yang, N. Bourbakis, and S. Li, "Data-image-video encryption," *Potentials, IEEE*, vol. 23, no. 3, pp. 28–34, aug.-sept. 2004.

[4] Y. Wang, K.-W. Wong, X. Liao, and G. Chen, "A new chaos-based fast image encryption algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 514–522, 2011.

[5] F. Sun, S. Liu, Z. Li, and Z. Lü, "A novel image encryption scheme based on spatial chaos map," *Chaos, Solitons & Fractals*, vol. 38, no. 3, pp. 631–640, 2008.

[6] H. Liu and X. Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Computers & Mathematics with Applications*, vol. 59, no. 10, pp. 3320–3327, 2010.

[7] P. Deepthi and P. Sathidevi, "Design, implementation and analysis of hardware efficient stream ciphers using lfsr based hash functions," *Computers & Security*, vol. 28, no. 3-4, pp. 229–241, 2009.

[8] S. Bojanic, G. Caffarena, S. Petrovic, and O. Nieto-Taladriz, "Fpga for pseudorandom generator cryptanalysis," *Microprocessors and Microsystems*, vol. 30, no. 2, pp. 63–71, 2006.

[9] I. Kokolakis, I. Andreadis, and P. Tsalides, "Comparison between cellular automata and linear feedback shift registers based pseudo-random number generators," *Microprocessors and Microsystems*, vol. 20, no. 10, pp. 643–658, 1997.

[10] B. Shackleford, M. Tanaka, R. J. Carter, and G. Snider, "Fpga implementation of neighborhood-of-four cellular automata random number generators," in *FPGA '02: Proceedings of the 2002 ACM/SIGDA tenth international symposium on Field-programmable gate arrays*. New York, NY, USA: ACM, 2002, pp. 106–112.

[11] M. Tomassini, M. Sipper, and M. Perrenoud, "On the generation of high-quality random numbers by two-dimensional cellular automata," *IEEE Transactions on Computers*, vol. 49, pp. 1146–1151, 2000.

[12] C. Torres-Huitzil, M. Delgadillo-Escobar, and M. Nuno-Maganda, "Comparison between 2D cellular automata based pseudorandom number generators," *IEICE ELECTRONICS EXPRESS*, vol. 9, no. 17, pp. 1391–1396, 2012.

[13] H. Kwok and W. K. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," *Chaos, Solitons & Fractals*, vol. 32, no. 4, pp. 1518–1529, 2007.

[14] M. Mohsen, G. Zied, Z. Medien, and T. Rached, "Design of reconfigurable image encryption processor using 2-d cellular automata generator," *IJCSA*, vol. 6, no. 4, pp. 43–62, 2009.

[15] H.-C. Chen and J.-C. Yen, "A new cryptography system and its vlsi realization," *Journal of Systems Architecture*, vol. 49, no. 7-9, pp. 355–367, 2003.