Proceedings of the 4th WSEAS Int. Conf. on Information Security, Communications and Computers, Tenerife, Spain, December 16-18, 2005 (pp240-245)

Design of a Parallel Computing Based Cryptosystem

MONICA DASCALU[1], EDUARD FRANTI[2], LUCIAN MILEA[3], TEOFIL TEODORESCU[3], TRAIAN BALAN[2], ANGELA IONITA[4]

[1] ICIA Bucharest, 13, *13 Septembrie Street*, Romania, monica@atlas.cpe.pub.ro;
[2] IMT Bucharest, 32, *Erou Iancu Nicolae Street*, Romania,
[3] Politehnica University of Bucharest, 1-3, *Iuliu Maniu Street*, Romania, +4-021-3169625;
[4] ICIA Bucharest, 13, *13 Septembrie Street*, Romania;

Abstract: - This paper presents a complex project that aims the development and design of a parallel computing based cryptographic system. Cryptography is a field of major scientific and technical interest nowadays, mainly because of the increased security demands of different communication systems. The novelty and the main interest in this project is the orientation towards the hardware implementation. Special software has been developed for the analysis of different encryption/decryption strategies with cellular automata. Between the various strategies and algorithms studied, the best were selected for hardware implementation. The main issues of cellular automata hardware design are also discussed for the selected cryptosystems. The result is a VLSI project for a fast, cheap, efficient and versatile cryptographic chip.

Keywords: cryptography, parallel computing, cellular automata, hardware/software co-design

1 Introduction

In this age of information, communications and electronic connectivity, security is a topic of general interest that should never be underestimated. The security of data bases, of data communications, of Internet connections, of scientific research and of personal e-mail and phone calls are some cases in which the encryption of data/information plays a major role. Therefore, cryptography has become an important field of theoretical research and applications development, not only in military communications as it was at its origins.

Because of its importance, cryptography is nowadays a science by itself, strongly related to other modern research fields as complexity theory, chaos, dynamical systems, computing theory etc. The state-of-the art for the field of cryptography is probably classified as it has military applications, but for the public domain a good reference can be found in [1] and [2]. The *encryption* of a message/data file/other information is a process (algorithm) that modifies this message/data file/information making it completely unintelligible, except for the person who knows the encryption *key*. The *key* refers to the encryption algorithm that has been used – in fact, to the reverse algorithm that should be used for decryption – and the particular parameters that have been used during the encryption. The *decryption* algorithm should render the original message/file/information complete and unaltered. Encryption can be achieved by constructing two different types of ciphers – stream ciphers and block ciphers. In the case of the block cipher, the message is split into successive blocks that are encrypted using a single key or multiple keys. In a stream cipher the message is broken into successive bits or characters and then the string of characters is encrypted using a key stream. The *cryptographic scheme* refers to the assembly of encryption and

decryption algorithms. An ideal cryptographic scheme or algorithm has not been developed yet, as an ideal cryptographic scheme implies: no data expansion during the encoding process; fast encoding algorithm; small dimension key; fast decoding algorithm; correct and complete rendering of message after encryption/decryption; invulnerability to attacks.

The last point is a major issue in cryptography; complex mathematical studies and research have to be done in order to establish the vulnerability of each cryptographic scheme. In simple words, this answers the basic question: how difficult is to break the code? This "difficulty" has to be established in terms of complexity, cost and computing time. Therefore, depending of the particular applications, sometimes it is enough to have a code and cryptographic scheme that requires a long search for the key, although the process is very simple. This is the situation for the briefcases with cipher, where the breaking process is quite simple: one has to try *all* possible numbers in order to find the right one. Cellular automata are applied with success in cryptography mainly because their vast phenomenology and apparently big complexity require a very long computing time to break well-chosen cryptographic schemes. There are indeed a lot of parameters and factors that can drastically affect the encrypted message (cyphertext) and therefore the complexity of the attack is considerably increased. Cellular automata offer an ideal mathematical model for massive parallel computation, but most research and applications in cellular automata domain are done through simulation. However, it is obvious that only the hardware implementations of this model fully exploit its computing and high-speed possibilities. In particular, cellular automata applications in cryptography are efficient because of the massive parallelism of the model. When implemented by means of other computing systems (simulated in software or emulated with microcontrollers etc.) the parallel processes are in fact executed sequentially. Special cellular automata hardware is the only means to benefit of all the advantages of the model. This is the main reason why CRIPTOCEL project was developed.

2 Cryptography with cellular automata

A. The Cellular Automata Model

Basically, cellular automata are parallel systems that consist of a typically large number of finite automata (finite state machines) as elementary "cells". The cells are locally connected, in other words the global network supports only local connections. The system evolves through local changes: all cells are updated synchronously, depending on their own current state and on the states of the neighbouring cells in the network. The computing task performed by cellular automata is generally conceived as the global evolution of its configuration, starting from the initial configuration (input data) and leading to an intermediate or final configuration or attraction cycle that are interpreted as a result. This almost "visual" perceptible computation is quite an advantage in tasks like modeling and simulation, image processing and cryptography. Cellular automata are defined by the following elements: topology and dimension of the cell lattice; the neighbours set of each cell that are involved in the next state's computation; the number of states for each cell (identical for all cells; binary automata, for instance, have only two states per cell: 0/1, visually translated as white/black); the local rule that gives the next cell state.

In cryptography, the main topologies that are applied are liner and two-dimensional, referring to a row or matrix of "cells".

B. Complexity of Cellular Automata

The cellular automata model is inspired from the natural model of complex systems that often consist of a large number of simple basic elements, having only local interactions that lead to a complex global behavior. This model is an important research and simulation tool in the science of complexity. Strictly speaking, the cellular automata model and its

evolution are not complex, as their structure and rules are very regular and simple. But the huge dimension of the rules and configuration spaces confers to the cellular automata phenomenology a considerably great *apparent* complexity. In the quite simple example of binary linear cellular automata with 100 cells (a modest dimension) with local rules involving the central cell and two neighbors on each side (binary functions with 5 variables), there are: 2^{100} global configurations, which represent ~10^{30} and 2^{32} possible local functions. In a rough approximation, this means around 4,000,000 functions.

Not all possible functions are of practical interest, but even in these conditions a search in the local functions' space is a very long computing task. This is related to the difficulty of cellular automata synthesis: there is no algorithm that gives the appropriate local function for a specific application. The universality of the cellular automata model is theoretically proven, but the practical applications are still waiting for development tools, since the experiment is, by now, the main means of cellular automata synthesis. The work of S. Wolfram [3] imposed a certain order in the space of local rules of functions, mainly for linear cellular automata, dividing it into four complexity classes (that express how complex is the evolution of the system governed by a certain rule). Class 3 of Wolfram's classification contains the automata that have an apparently chaotic evolution, also strongly depending on the initial global configuration. Such an evolution is illustrated in Fig. 1, where time is on the vertical axis, downwards, and the horizontal bit string is the configuration of a binary linear cellular automata.

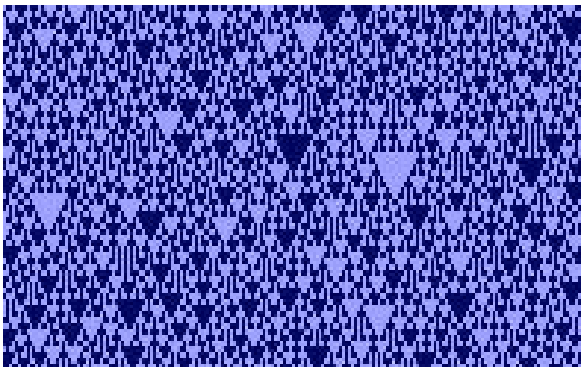

Fig. 1. Evolution of class 3 linear cellular

The *class 3 automata* are ideal for applications like random sequence generation and cryptography [3],[4].

C. Applications in Cryptography

The very large phenomenology of the cellular automata model and its apparently big complexity offer a good basis for applications in cryptography (cellular automata are not the only dynamical systems applied in cryptography, and some of the basic principles of cryptography with cellular automata also stand for other dynamical systems). Massive parallelism is another feature of cellular automata that make this model attractive for cryptography, since lot of computation is often necessary in real-time applications. However, this parallelism, when emulated in software or in sequential hardware, disappears. In the last two decades, many results have been obtained in this direction. A good review of some cryptographic schemes and systems with cellular automata proposed in the scientific literature is given in [5] and [6]. In the next section some of these will be discussed as possible basis for hardware implementations, taking as valuable the theoretical results concerning the efficiency and invulnerability of the schemes.

D. Other Advantages of Cellular Automata

The design style under VLSI technology prefers the simple, modular and local connected logic circuit structure. Cellular automata are ideal from the hardware designer's point of view. The local connectivity, regularity and the simple basic components make CA very appropriate to implement low-cost and robust massive parallel machines or application-oriented circuits. The reason why there are not so many cellular-automata inspired electronic circuits is mainly the difficulty of synthesis for this computing model; however, dedicated circuits for specific applications (signal generators, associative memories, image pre-processing blocks, various simulators for natural phenomena) have been successfully designed and produced [6].

3 Selection of algorithms

A. CRIPTOCEL Software

A special software tool was developed in order to select the cryptographic schemes that are optimal for hardware implementation. This software is necessary mainly in order to choose a (large) set of local rules as possible keys that should be integrated in hardware. The software accepts linear and two-dimensional cellular automata having variable dimensions, limited to 100/100 cells (that gives a maximum of 10^4 bits for two-dimensional encryption algorithms). In addition to the cellular automata simulation, the software completely emulates the algorithms for encryption and decryption. The software is an efficient tool for empirical analysis as well as statistic studies of all stages of the cryptographic schemes. An attractive software facility is the computation of the map of the basins of attractions, which stands only for small dimension linear cellular automata but can be qualitatively extrapolated for larger systems having the same rule.

B. Simulation of Cryptographic Schemes

We have studied four types of cellular automata-based cryptosystems.

(1) The first type of cryptosystem uses (typically) *linear class 3 cellular automata* as a pseudo-random generator that generates a key stream to be combined with the original message. This is split in words of given length, equal to the number of cells in the cellular automata. The key is the combination of the local rule and initial configuration. At the starting point of the encryption process, the cellular automata evolution starts. Step by step, the current configuration is a new binary word. The encryption is done through a logical combination (typically, bitwise XOR function) of each word of the original text with each configuration of the cellular automata. At each moment (time step) the encoded word of the cyphertext may be further transmitted. The decryption shall be done in the same manner, as a double application of XOR function restores the original data. Therefore, if someone knows the topology, the local rule and the initial configuration of the cellular automata, the original message can be easily obtained. The so-called “rule 30” with 3 neighbours is a typical class 3 function, but it is not the only one. The simulations have proved that the *seed* (initial configuration) if not well-chosen may affect the behavior of the cellular automata which may be too simple and does not alter enough the plaintext. Results obtained: a library of local rules for different neighbourhoods and seeds to be transposed in hardware. This scheme is most appropriate for hardware implementation.

(2) Reversible cellular automata offer a very simple cryptographic scheme. The initial message (or a block of it) is the original configuration of a reversible cellular automata. The encryption algorithm is based on the forward evolution of the system, and the cyphertext is an arbitrary chosen resulting state, after a number of steps that ensure the complete meaningless of the global configuration.

The decryption is done by reverse running of the given cellular automata – having the cyphertext as initial state – for the same number of steps. This is a very elegant cryptosystem, but lacks theoretical support if two-dimensional cellular automata are used [4]. There are quite few proven reversible laws in the two-dimensional functions’ space, and therefore the key (which is basically the evolution rule) is easy to discover if the scheme and the topology are known. Results obtained: analysis of reversible local laws in linear cellular automata.

(3) Backward iteration of irreversible cellular automata is a much more effective encryption strategy. In this type of cryptographic scheme, usually applied for two-dimensional cellular automata, the original message is the starting configuration. Instead of normal evolution, the system is run backwards. One of the possible predecessors of the current state is chosen in a nondeterministic way for the reverse trajectory. The large size of the automaton, combined with proper

local laws may ensure a map of attraction basins with long ramifications and quite few “garden of Eden” states, an important condition for this algorithm. The decryption of the cyphertext is simply done through direct iteration of the local law, which is the encoding key. This is a very attractive scheme, but the hardware transposition of the encryption algorithm is utopian, because of the backward iteration that is usually computed with specific software. If the encryption is done by a computer, the decryption circuit in its hardware version is quite simple.

(4) Finally, the last cryptographic scheme analyzed uses a *set of local laws* that are applied in a given sequence during the encryption process. Each rule may be applied for a different number of time steps, in order to introduce as much as possible unknown variables for a possible attack. The receiver of the message has to know the right sequence of functions and the number of steps that each function is applied. This complicate evolution of the cellular automata can be used in two manners: as in the first type of scheme discussed, we may consider this as a key stream to be combined with the plaintext. Or, as in the third type, we can run backwards the system with the combination of laws for encryption and then run it forward with the inverse sequence for decryption. In this last case, the message is considered as the start configuration and the discussion concerning the hardware implementation is the same. The first idea (with key stream) is appropriate for hardware version. Results obtained: a library of local rules for two-dimensional cellular automata that are not fast convergent to attractor cycles starting from random seeds. For linear topology, also of interest for this scheme, analysis was focused on large neighbourhoods.

C. Discussion

In the field of two-dimensional cellular automata or linear cellular automata with large dimension of the neighbourhood the exhaustive analysis is not possible, for reasons stated in section II. The process is somehow random, implying the observation of the global state evolution for different rules and seeds and eventually the statistic analysis of some chosen rules. An important step of the algorithms must be emphasized here: the original message is usually divided in blocks or words that are encoded and transmitted sequentially. Therefore, the dimension of the blocks or words, which is directly related to the dimension of the cellular automata, is another parameter that can be modified in some limits, improving the invulnerability of each of the schemes.

4 Hardware implementation

The hardware implementation that is the final goal of the CRIPTOCEL project is quite easy to design. We present here the block scheme for the key stream encryption algorithms (of type 1 and 4 in the list above). This block scheme include (Fig. 2) a central unit (CU) (that generates the configuration signals for the word dimension, the synchronization signals and the rules for the cellular automata), the cellular automata (CA) lattice that occupies most of the chip area, the block generator (BG) that divides the original data stream in blocks or words of given dimension and a logic bitwise XOR (exclusive-or) circuit.

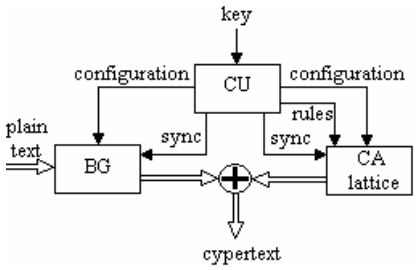

Fig.2 The block scheme for the key stream encryption

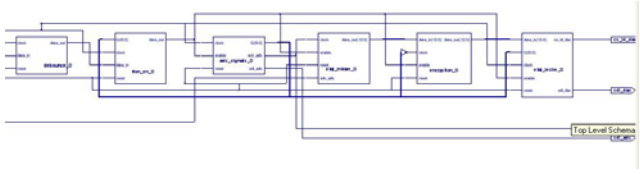

Fig.3 The cryptographic block scheme

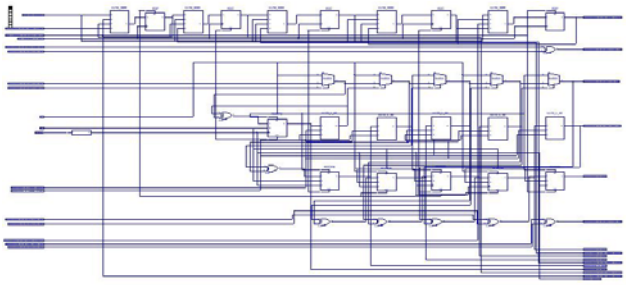

Fig.4 The synthesis of the cryptographic block in Xilinx technology

The cellular automata lattice is a two-dimensional network of finite state machines. Even for linear topology, the cells are geometrically arranged in two-dimension and connected to obtain the linear network. The control and coordination of such a cellular automata lattice is presented in [7]. The basic cell is presented in Fig. 3, having a state register R and a programmable logic circuit.

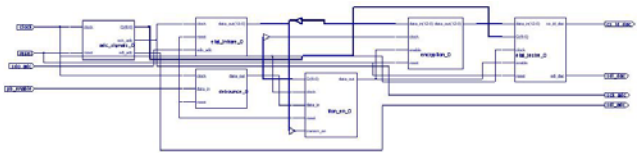

Fig.3 The decryption block scheme

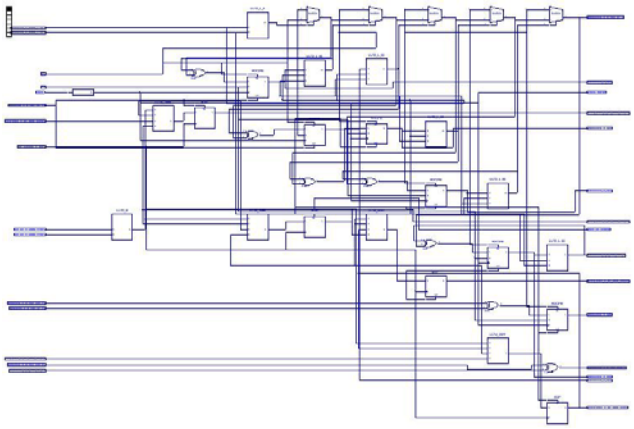

Fig.4 The synthesis of the decryption block in Xilinx technology

Note that the neighborhood dimension is a fundamental parameter for the hardware implementation of the local logic circuits, as the dimension of this circuitry depends exponentially on the neighborhood dimension.

Starting from this general configuration, many simple particular cryptography tasks may be implemented for different specific products, each of them having a specific set of rules to be transposed in the PLC.

5 Conclusions

Cipher schemes with cellular automata have already been studied and proved to be efficient and having good invulnerability properties. The paper presents the development and design of specific hardware, using a dedicated simulator to emulate various cryptographic schemes with different parameters. The VLSI implementation will result in cheap, robust, efficient and versatile encryption and decryption circuits that may be adapted to different devices (like phones) in order to ensure the privacy of data communication.

Acknowledgment:

The work of this paper was done with support from SCRIPT 8/2005 project from the SECURITATE Romanian Research Program

References:

[1] William Stallings, *Cryptography and Network Security* Prentice-Hall, New Jersey, USA, 2003.

[2] Brian A. LaMacchia, *.NET Framework Security*, Addison Wesley, USA, 2002.

[3] S.Wolfram, *Cellular Automata and Complexity*, Addison-Wesley Publishing Company, 1994.

[4] J. Kari “Reversibility of 2D Cellular Automata is Undecidable”, *Physica D,* Vol. 45, pp. 379-385, 1990.

[5] H.Gutowitz, *Method and Apparatus for Encryption, Decryption and Authentication using Dynamical Systems* US Patent 5365589, 1994.

[6] P. Chaudhuri, D. Chowdhury, S. Nandi and S. Chattopadhyay, *Additive Cellular Automata: Theory and Applications*, IEEE Computer Society Press, 1997.