An Efficient Implementation of Cryptographic Algorithm Using High Speed Cellular Automata Techniques

G. S. Khedkar*, Prof. A.O. Amalkar**, S.S.Tawani***

* (Department of Electronics & Telecommunication Engineering
SSGMCE, Shegaon, Dist. Buldhana.)

** (Department of Electronics & Telecommunication Engineering
SSGMCE, Shegaon, Dist. Buldhana.)

*** (Department of Electronics & Telecommunication Engineering
Sipna C.O.E.T., Amravati.)

ABSTRACT

With the ever-increasing growth of data communication, the need for security and privacy has become a strong necessity. In these conditions, the necessity of new powerful encryption techniques becomes a crucial issue. In this paper Cellular Automata (CA) are applied to construct cryptography algorithms. The cryptography applications in the epoch of the informational society are practically unlimited. In our information age, the need for protecting information is more pronounced than ever. Secure communication for the sensitive information is not only compelling for military and government institutions but also for the business sector and private individuals. Modern telecommunication networks, and especially the Internet and mobile-phones networks, have tremendously extended the limits and possibilities of communications and information transmissions. Associated with this rapid development, there is a growing demand of cryptographic techniques, which has spurred a great deal of intensive research activities in study of cryptography. There is not a method of ideal codification at present and this is an additional reason regarding the suitability of the elaboration of new methods and encryption systems. Cryptography is the best solution against the unauthorized use of the information. Cryptography is a permanent field of interest at all times. The purpose of the cryptography is to hide the contents of messages by encryption them, so as to make them unrecognizable except by someone who has been given a special decryption key. The purpose of crypto-analysis is then to defeat this by finding ways to decrypt messages without being given the key.

Keywords **– Cellular Automata, Cryptography, Crypto analysis, Decryption, Encryption.**

I. INTRODUCTION

Cryptography is an important and vital application in security, defence, health, business and many other application areas. Cryptographic techniques are divided into two categories: symmetric-key (or secret key) and asymmetric-key (or public-key). The difference is that symmetric algorithms use the same key for encryption and decryption (or the decryption key is easily derived from the encryption key), whereas asymmetric algorithms use a different key for encryption and decryption, and the decryption key cannot be derived from the encryption key without knowing some additional secret information.[1]

There are two classes of symmetric-key encryption schemes: block ciphers and stream ciphers. Block ciphers breaks up the message into blocks of the fixed length and encrypt one block at a time. Stream ciphers can encrypt a single bit of plain text at a time. Good overview of all major cryptography technique can be found in. This paper deals with symmetric-key block encryption. CA have been used so far in both symmetric-key and public-key cryptography. Our goal is to develop an alternative cryptogram based on hybrid cellular automaton (CA), in which several CA technologies such as Wolfram approach [2], transform-based approach [3] and five evolution rules are combined in some way to form a one cryptogram.

This paper present a hardware implementation in a FPGA circuit of an efficient encryption algorithm based on Cellular Automata. We demonstrate in this thesis how microscopic (local) interactions influence the overall macroscopic (global) behavior of the whole system. A regular, modular, and cascadable hardware implementation of the encryption system has been implemented, which is efficient in terms of VLSI technology. The experimental results prove the powerful of cellular automata encryption systems. The method supports both software and hardware implementation. The design has been specified in VHDL targeted on a XC3S400 based FPGA and can be verified for functional correctness by software simulation.[2]

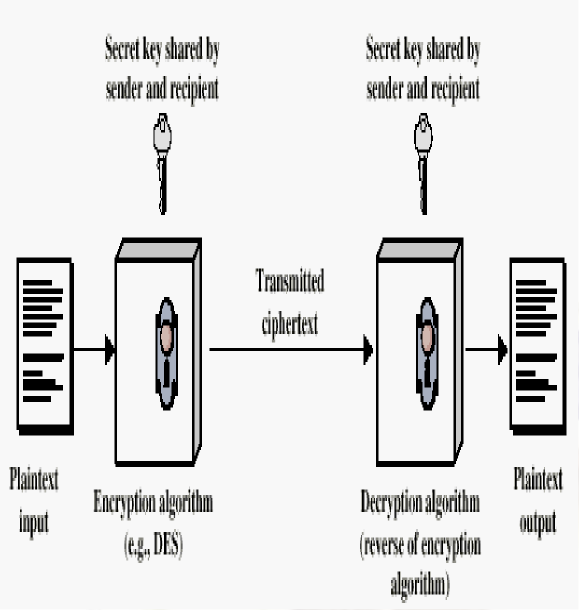

Secret key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted ciphertext
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Fig. 1 Simplified Model of Conventional Encryption

Original message/data fed as i/p to algorithm is known as Plaintext. The process or algorithm in which various substitutions and transformation are obtained on the plain text is called as Encryption algorithm. Substitutions and transformation depends on this key and the secure information require to obtained encryption and encryption is called as Secret Key. Scrambled msg. produced as o/p i.e. unreadable form of data is called as Cipher text. This is nothing but a encryption algorithm which runs in reverse i.e. the process of obtaining the plain text from cipher text is called as Decryption algorithm [3].

1.1 CA CONCEPTS

As concept of cellular automata (CA) - cellular spaces [4], to the recent book of Stephen Wolfram "A New Kind of Science" [2], the simple structure of CA has attracted researchers from various disciplines. CA is a bio-inspired paradigm highly addressing the soft computing and hardware for a large class of applications including information security. They are a particular class of dynamical systems that enable to describe the evolution of complex systems with simple rules, without using partial differential equations [5]. Typically, a cellular automaton consists of a graph where each node is a finite state automaton (FSA) or cell. This graph is usually in the form of two-dimensional lattice whose cells evolve according to a global update function applied uniformly over all the cells.

The state of each cell is updated simultaneously at discrete time steps, based on the states in its neighborhood at the preceding time step. The algorithm used to compute the next cell state is referred to as the CA local rule. If the rule of a CA involves only XOR logic, then it is called a linear rule. Rules involving XNOR logic are referred to as complement rules.[4] A CA with all the cells having linear rules is called a linear CA, whereas a CA having a combination of XOR and XNOR rules is called additive CA. If all the cells obey the same rule, then the CA is said to be a uniform CA, otherwise, it is a hybrid CA. A CA is said to be a null boundary CA, if both the left and right neighbor of the leftmost and rightmost terminal cell is connected to logic 0-state.

1.2 CA STRUCTURE

This update function takes the cell's present state and the states of the cells in its interaction neighborhood. The cells evolve in discrete time steps according to some deterministic rule that depends only on local neighbors. Each cell consists of a storage element (D flip-flop) and a combinational logic (CL) implemented the next-state functions. A huge flexibility into this programmable structure can be introduced via control signals in CL. For an n-cell CA structure can be used for implementing 2n CA configurations. [5]

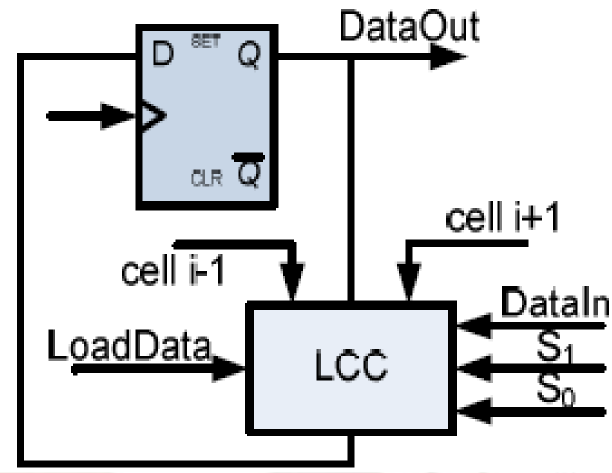


Fig. 2 Shows detail schematic of CA structure

Cellular automata are a metaphor of a universe whose physics is reduced to some fundamental simple laws. Eight cells are connected together to form an 8-cell CA. In the 8-cell CA, the data path is 8 bits data(0-7), there are 9 rules configuration 511 A signals (Rule[0:8]) and one load data control signal (Load Data). Because all the cells share the same common rule signal SO, only rule 51 or one of the rules 153 or 195 can be applied at a given time. The common rule configuration signal (Rule8) simplifies the LCC. Every cell has its own rule configuration signal (RuleO-Rule7). The left and right input terminals of every cell are connected to left and right neighbors Data Out terminals, thus it is configured as a 3- neighborhood CA. The left and right terminals of the leftmost and rightmost cells are connected to "logic 0" providing a null boundary condition [6].

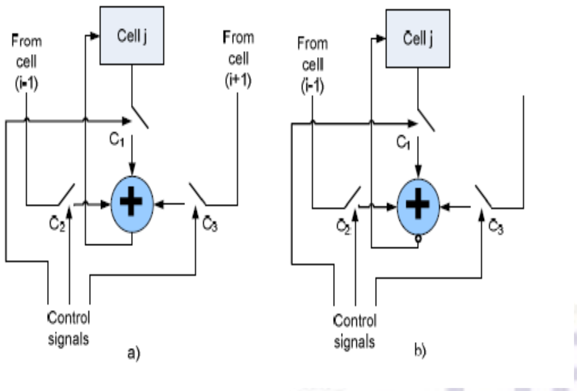


Fig. 3 Programmable CA
a)Non complemented Rule b) Complemented Additive Rule

II. CONCEPT OF CA RULES

Automata can be seen in multiple fields of science and social science. They range from computer science, technology, physics, biology, and math, to economics, psychology, philosophy, and even art. The research we have focused on applies mainly to cryptology and random number generation. Before any further mention of randomness, a definition must be established. The concept of "random" can be described in a variety of ways. The definition we have referred to in our research can be credited to Dr. Solomon Golomb, a professor at the University of Southern California. Dr. Golomb proposed three postulates for randomness, which he classified as preliminary steps. If a binary string passes all three postulates, the string can be considered as pseudo-noise and qualifies for further inspection [7].

S1	S0	RULE APPLIED
0	0	30
0	1	51
1	0	153
1	1	195

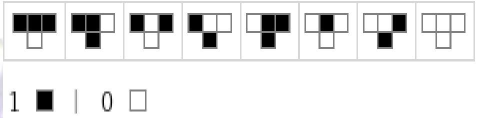
Table 1: Selection of Rules 30, 51, 153, 195

2.1 RULE 30

Dr. Stephen Wolfram, developer of Mathematica, claims that Rule 30 can be used as an effective encryption scheme due to its random qualities. We investigate this claim by using a battery of statistical tests as well as identify properties that help characterize its security if used for encryption. The results provide evidence that Rule 30 shows adequate randomness for a high level of security with weaknesses isolated to even window sizes.

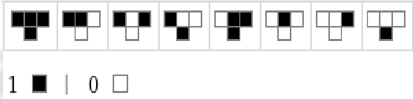
BOOLEAN: P XOR (q OR r)
ALGEBRIC FORM: (p + q + r + qr) mod2

2.2 RULE 90



BOOLEAN FORM: F (P, q, r) = p XOR r
ALGEBRIC FORM: F (p, q, r) = (p + q) mod2

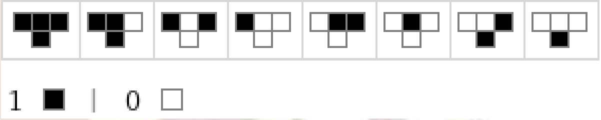
2.3 RULE 153



BOOLEAN FORM: F(p, q , r)=NOT(p XOR q)
ALGEBRIC FORM: F(p, q ,r) = (1+p+q) mod2

2.4 RULE 195

BOOLEAN FORM:
F(p, q ,r)=NOT (q XOR r)
ALGEBRIC FORM: F (p ,q ,r) = (1+p+r) mod2



III. CA ENCRYPTION ALGORITHM

The encryption method proposed in this thesis is based on the fact that the CAs from class III are chaotic dynamical systems and CAs from class II exhibit periodic behavior. In these cases, their evolution depends essentially of the initial state, but we can say that after a while the initial state is "forgotten", in sense that the initial state cannot be retrievable through analyses of the current configuration. The proposed encryption system it is realized using a combination of two CA. We use a first CA as a key stream generator, a CA pseudorandom number generator (PSRG) that combines in some way two rules (the rules 90 and 150), to provide the key sequence. The rules 90 and 150 can be expressed as follows:

$$a_i(t+1) = a_{i-1}(t) \oplus a_{i+1}(t) \qquad \textit{Rule 90}$$

$$a_i(t+1) = a_{i-1}(t) \oplus a_i(t) \oplus a_{i+1}(t) \quad \textit{R. 150}$$

This CA is used to provide real-time keys for the block cipher in this thesis. The operation of CA can be represented by a state-transition graph. Each node of the transition graph represents one of the possible states of the CA. The direct edges of the graph correspond to a single time step transition of the automata. Fig.4 shows the state transitions graph of a 4-bit hybrid null boundary condition CA with rules <90, 150, 90, and 150> [8].

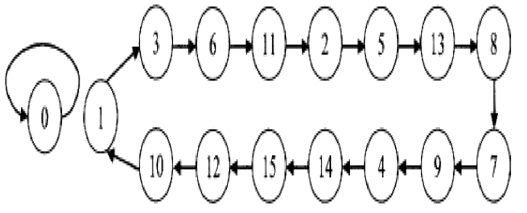


Fig. 4 The State Transitions Diagram of A Maximum-Length Group CA.

Eight cells are connected together to form an 8-cell CA. In the 8-cell CA, the data path is 8 bits (Data[O: 7]), there are 9 rules configuration 511 A signals (Rule[0:8]) and one load data control signal (Load Data). Because all the cells share the same common rule signal SO, only rule 51 or one of the rules 153 or 195 can be applied at a given time. The common rule configuration signal (Rule8) simplifies the LCC. Every cell has its own rule configuration signal (RuleO-Rule7). The left and right input terminals of every cell are connected to left and right neighbors Data Out terminals, thus it is configured as a 3- neighborhood CA. The left and right terminals of the leftmost and rightmost cells are connected to "logic 0" providing a null boundary condition [9].

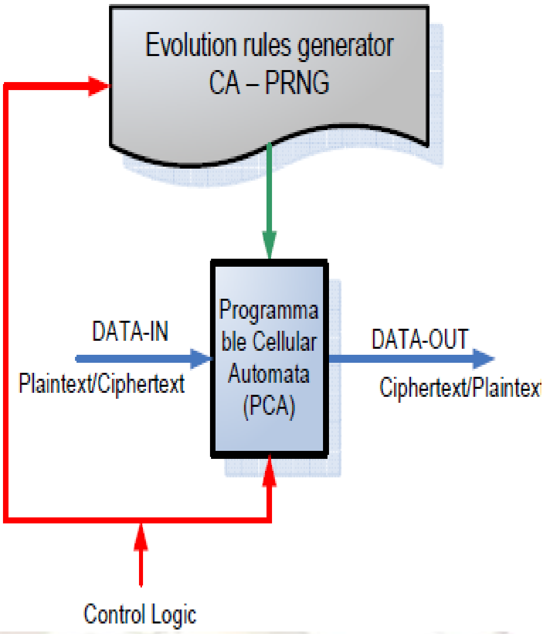
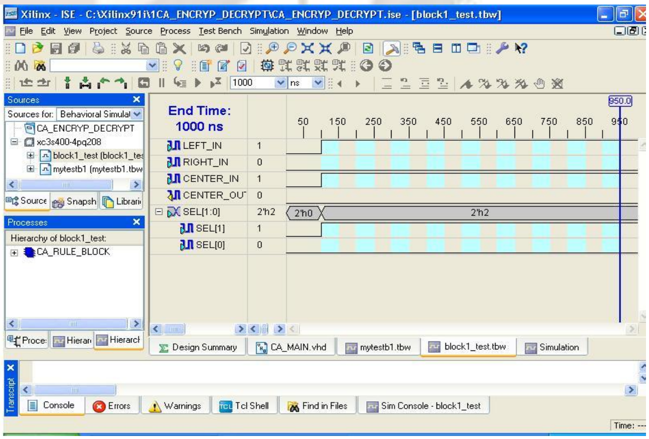


Fig. 5 Encryption System Structure.

STEPS

1. Load the PCA with one byte plaintext (cipher text) from I/O. The initial block of the message is the initial state of the PCA. The global configuration of the PCA represents the encrypted message.
2. Load a rule configuration control word from memory into the PCA.
3. Run the PCA for 1 … 7 cycles.
4. Send one byte cipher text (plaintext) to I/O.
5. If not end of the plaintext (cipher text) go to step 1. Otherwise, stop the process.

IV. RESULTS



Xilinx - ISE - C:\Xilinx91i\1CA_ENCRYP_DECRYPT\CA_ENCRYP_DECRYPT.ise - [block1_test.tbw]

Sources for: Behavioral Simulat

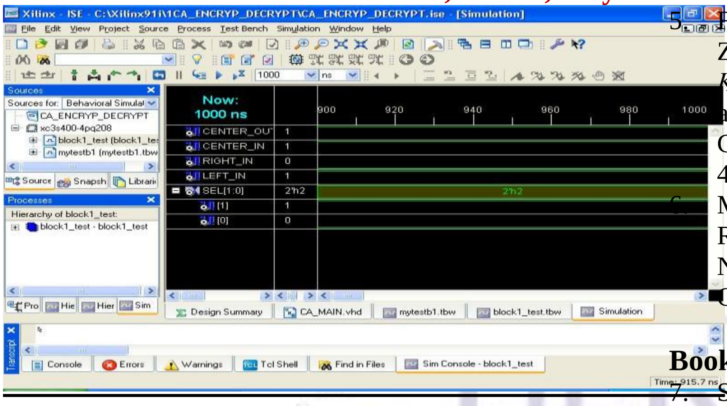
CA_ENCRYP_DECRYPT
xc3s400-4pq208
block1_test (block1_tes
mytestb1 (mytestb1.tbw

Hierarchy of block1_test:
CA_RULE_BLOCK

End Time: 1000 ns

LEFT_IN	1
RIGHT_IN	0
CENTER_IN	1
CENTER_OU	0
SEL[1:0]	2'h2
SEL[1]	1
SEL[0]	0

Fig. 6 Waveforms for CA Rule Before Simulation.


Xilinx - ISE - C:\Xilinx91i\1CA_ENCRYP_DECRYPT\CA_ENCRYP_DECRYPT.ise - [Simulation]
File Edit View Project Source Process Test Bench Simulation Window Help
1000 ns
Sources
Sources for: Behavioral Simulat
CA_ENCRYP_DECRYPT
xc3s400-4pq208
block1_test (block1_te
mytestb1 (mytestb1.tbw
Source Snapsh Librari
Processes
Hierarchy of block1_test:
block1_test - block1_test
Pro Hie Hier Sim
Now:
1000 ns
900
920
940
960
980
1000
2'h2
Design Summary
CA_MAIN.vhd
mytestb1.tbw
block1_test.tbw
Simulation
Transcript
Console Errors Warnings Tcl Shell Find in Files Sim Console - block1_test
Time: 915.7 ns

```
CENTER_OU'   1
CENTER_IN    1
RIGHT_IN     0
LEFT_IN      1
SEL[1:0]     2'h2
  [1]        1
  [0]        0
```

Fig. 7 Waveforms For CA Rule After Simulation.

V. CONCLUSION

This system has tried to demonstrate the efficiency of the hardware implementation of a CA based encryption algorithm. The method involved can compare the results obtained by a FPGA implementation of CA with the PC software simulation. General accepted expectations were confirmed regarding the following aspects: - In the CA encryption algorithm, the same cipher text may be generated from different plaintext, and any cipher text may give rise as well to different plaintext under different CA rule configurations. Thus, the scheme is guarded against the cryptanalyst's cipher text only attack. - For higher security system, 32- or 64- or 128- bit block size would be more appropriate. In terms of the hardware efficiency we consider the valuable results in this work as follows:
The implementation requires small area of silicon because of the low number of gates involved in cryptography application, the presented solution has proved a high speed of information processing in terms of random number generating , this approach creates large possibilities to implement efficient encryption systems, path and the maximum operating frequency.

REFERENCES

Journal Papers:

1. Niloy Ganguly, Biplab Sikdar, Andreas Deutsch, Geofrey Canright, Pal Chaudhuri,” A Survey on Cellular Automata”, Project BISON, IST-2001-38923.
2. F. Schweitzer, J. Zimmermann,” Communication and Self-Organization in Complex Systems”: A Basic Approach, Knowledge, Complexity and Innovation Systems, pp. 275-296, 2001
3. P. D. Hortensius, R. D. McLeod, Podaima, “Cellular automata circuits for built-in-self-test”, IBM J. RES. DEVELOP., Vol. 34, No. 2/3, pp. 389-405, 1990.
4. S. Nandi, B.K. Kar, P. Pal Chaudhuri, “*Theory and applications of cellular automata in cryptography*”, IEEE Transactions on Computers, **43**(12), 1994, pp. 1346-1356.
5. Franciszek Seredynski, Pascal Bouvry, Albert Y. Zomaya, *“Cellular Programming and Symmetric Key Cryptography Systems”,* in GECCO 03 (Genetic and Evolutionary Computation Conference), Chicago, IL, USA, Springer 2003, ISBN 3-540-40603-4, pp. 1369-1381
6. M. Seredinsky, P. Bouvry, Block Encryption Using Reversible Cellular Automata, ACRI 2004 The Netherlands - Amsterdam, LNCS 3305, pp. 785-792, October 2004.

Books:

7. S. Wolfram, A New Kind ofScience, Wolfram Media Inc, 2002.
8. O. Lafe, Cellular Automata Transforms: Theory and Applications in Multimedia Compression, Encrypt and Modeling, Kluwer Academic Publisher, 2000.
9. R. Espericueta, “Cellular automata dynamics”, unpublished book, available at address: http://www2.bc.cc.ca.us/resperic/ca/, 1997.
10. C. Shannon, “*Communication Theory of Secrecy Systems*”, Bell Sys. Tech. J. 28, pp. 656-715, 1949
11. S. Wolfram, “Theory and applications of cellular automata”, *Wolfram Scientific*, 1986.
12. J. Von Neumann, “Theory of Self-Reproducing Automata”, University of Illinois, Urbana, 1966.

Thesis:

13. P. Anghelescu, “The projection and the analyses of the cellular automata for processing of information”, Doctoral Thesis (in Romanian), University of Pitesti, 2007. The summary is available on: http://www.upit.ro/index.php?i=20.