Stream Cipher Design using Elementary Cellular Automaton and its FPGA Implementation

V. Divya
Dept. of Electronics and Communication Engineering
Tagore Engineering College,
Chennai

T. Suresh M.E,
Asst. Professor
Dept. of Electronics and Communication Engineering
Tagore Engineering College, Chennai

***Abstract*—Pseudo-random number generators (PRNGs) are a key component of stream ciphers used for encryption purposes. Non-Linear Feedback Shift Registers (NLFSRs) have been proposed as an alternative to Linear Feedback Shift Registers (LFSRs) for generating pseudo-random sequences for stream ciphers. While linear feedback shift registers (LFSRs) combined with nonlinear feedback shift registers (NFSRs) have typically been utilized for PRNGs, but the use of cellular automata (CA) along with this registers increases the randomness. Stream ciphering devices seem to be one of the best alternatives in order to provide confidentiality to high-speed transmissions. This paper explores the combination of LFSRs and CA as the key components of an efficient stream cipher design for implementation on Field Programmable Gate Arrays (FPGAs). The proposed stream cipher design builds upon a recent published design known as A2U2. Stream cipher design overcomes the drawback of A2U2 design such as adjacent bit correlations and the appearance of repetitive structures in the bit sequences. The use of CA have the potential to improve the quality of the random numbers generated and hence increase the security of the cipher. Such developments bring new challenges, especially in terms of providing security, both to protect privacy as well as to enable applications dependent on security, such as e-tickets and e-banking.**

Keywords: Cellular Automata, Linear/Non- linear feedback shift registers, Stream Ciphers.

I INTRODUCTION

Cryptography is the practice and study of techniques for secure communication in the presence of third parties. More generally, it is about constructing and analyzing protocols that overcome the influence of third parties and that are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce. Cryptography referred almost exclusively to encryption, which is the process of converting ordinary information (called plaintext) into non meaningful text (called ciphertext). Decryption is the reverse, in other words, converting the non meaningful ciphertext back to plaintext. A cipher (or cypher) is a pair of algorithms that create the encryption and the decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a "key". This is a secret text (ideally known only to the communicants), usually a short string of characters, which is needed to decrypt the ciphertext. A "cryptosystem" is the ordered list of elements of finite possible plaintexts, finite possible ciphertexts, finite possible keys, and the encryption and decryption algorithms which correspond to each key. Keys are important, as ciphers without variable keys can be cryptanalyzed with only the knowledge of the cipher used and are therefore useless for most purposes.

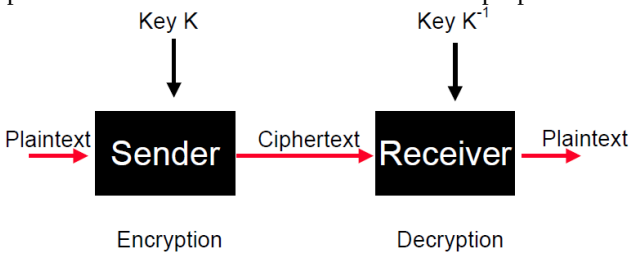

Figure 1: Basic Cryptography

A stream cipher is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (keystream). In a stream cipher each plaintext digit is encrypted one at a time with the corresponding digit of the keystream, to give a digit of the ciphertext stream. An alternative name is a state cipher, as the encryption of each digit is dependent only on the current state. In practice, a digit is typically a bit and the combining operation is exclusive-or (xor). The pseudorandom keystream is typically generated serially from a random seed value using shift registers. The seed value serves as the cryptographic key for decrypting the ciphertext stream. One of the main component of the stream cipher is the key stream generator, which can be viewed as a pseudo-random number generator (PRNG). Important metrics for the performance of the PRNGs are power dissipation, speed and area while producing high-quality random numbers. A linear feedback shift register (LFSR), which is implemented from a series of flip-flops and a few XOR gates, typically forms the core of a PRNG. In addition, nonlinear feedback shift registers (NFSRs) must be included in a key stream generator design to improve the non-linearity in the encrypted cipher text, making it more difficult for the third parties to discover the secret key.

In this paper, the impact of adding cellular automata (CA) to the keystream generator is considered. The implementation of CA is relatively simple using Field Programmable Gate Arrays (FPGAs) due to their nearest neighbor interconnectivity and regularity in their physical layout .The main contributions of this paper are the introduction of CA into stream cipher design, FPGA implementation and test of the stream cipher, and characterization results with DIEHARD and the entropy test.

This paper is organized as follows. The second section provides some previous work in the area of stream cipher design is also covered. The next section describes the research method used to design and test the stream ciphers. The fourth section contains the results and a discussion of their implication. Conclusions and future work are given in the final sections.

II EXISTING WORK

The design of hardware-oriented ciphers has an increasingly important role to play with emerging ubiquitous and pervasive computing devices, such as low cost passive Radio Frequency Identification (RFID) tags. The importance of such ciphers are further highlighted by novel manufacturing technologies, such as printed ink to develop extremely low cost RFID tags. Such developments bring new challenges, especially in terms of providing security, both to protect privacy as well as to enable applications dependent on security, such as e-tickets. In this proposed work, we develop a new stream cipher which is based on A2U2 design [1], which uses the approach block cipher design.

Linear Feedback Shift Registers (LFSRs) and Cellular Automata (CA) are most commonly used in the implementation of pseudo-random number generators (PRNGs) [4]. But, these designs typically cannot produce reliably a high quality random numbers due to adjacent bit correlations and the repetitive structures in the bit sequences. This paper explores the implementation of an efficient hybrid configuration which combines the bit streams from an LFSR and a CA. The proposed configurations takes the advantage of the FPGA's ability to realize compact LFSR implementations. Site spacing is utilized to reduce the effect of adjacent bit correlations and hence improve the pseudo-random number quality.

III DESIGN PRINCIPLES

The proposed stream is based upon the A2U2 stream cipher. The A2U2 stream cipher is a hardware-based stream cipher proposed for extremely resource limited devices such as RFID tags. The proposed stream cipher combines the A2U2's design principle with the elementary Cellular Automata. It consists of a 17 bit NFSR as the A2U2 stream cipher but replaces the shorter-length NFSR of the A2U2 with a 9 bit maximal length cellular automata as depicted in Figure 4.

The proposed stream cipher contains five different blocks. The five blocks are: i) a counter, ii) a 17 bit non-linear feedback shift register, iii) a 9 bit cellular automata, iv) a key bit mixing mechanism, and v) a filter function. The architecture of the proposed stream cipher is shown in Figure 2.

The first and foremost module in the proposed stream cipher is the counter (see Figure 3). The counter is based on LFSR as in the A2U2 stream cipher design because it has already been used in the design to reduce the number of gates.

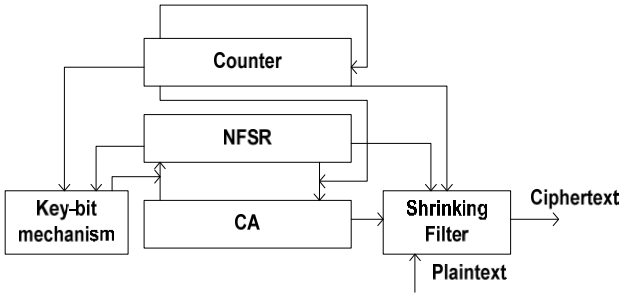

Figure 2:: Stream cipher Architecture

The feedback function is a maximal length polynomial function F_C (whose period is 2^7-1) defined by:

$$F_C = X^7 + X^4 + 1$$

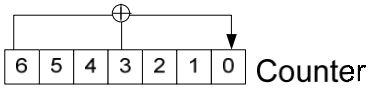

Figure 3: Counter used in proposed stream cipher

The next module is the interconnection between the NFSR and CA, the feedback of the NFSR provides the feedback to the CA and vice versa, as shown in Figure 4.

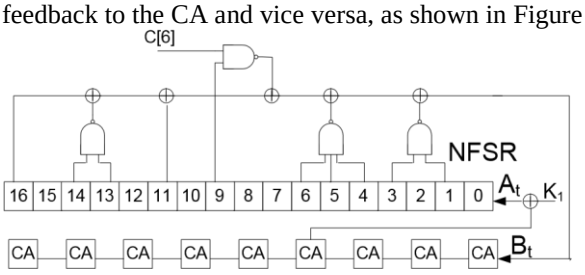

Figure 4: NFSR and the CA combination

An important core of the proposed stream cipher design is the implementation of CA. A cellular automata with binary state values can be viewed as an array of cells where each cell can assume either the value 0 or 1.Cellular automata (CA) are discrete, abstract computational systems that have proved useful both as general models of complexity and as more specific representations of non-linear dynamics in a variety of scientific fields. Firstly, CA are (typically) spatially and temporally discrete: they are composed of a finite or denumerable set of homogeneous, simple units, the atoms or cells. At each time unit, the cells instantiate one of a finite set of states. They evolve in parallel at discrete time steps, following state update functions or dynamical transition rules: the update of a cell state obtains by taking into account the states of cells in its local neighborhood.

Secondly, CA are abstract, as they can be specified in purely mathematical terms and implemented in physical structures. Thirdly, CA are computational systems: they can compute functions and solve algorithmic problems.

The rules start with the case when all the cells in the bottom row are white, and end with the case when they are all black. Therefore, there can be 256 possible combinations of black and white cells in the bottom row. In total, we can have 256 cellular automata elementary rules. If we replace the white and black cells with zeroes and ones respectively, then for the bottom row we receive the combinations starting from 00000000 (which means that all the cells in the bottom row are white) and ending with 11111111 (which means that all the cells in the bottom row are black). As one can recognize, these combinations of zeros and ones are nothing but the binary representations of the numbers 0 and 255 correspondingly. We will call Rule 000th initial combination, and Rule 255 the ending combination.

RULE 90

Rule 90 is an elementary cellular automaton based on the exclusive or function. It consists of a one-dimensional array of cells, each of which can hold either a 0 or a 1 value; in each time step all values are simultaneously replaced by the exclusive or of the two neighboring values.When started from a random initial configuration, its configuration remains random at each time step; however, any configuration with only finitely many nonzero cells becomes a replicator that eventually fills all of the cells with copies of itself. An assignment of values to all of the cells is called a configuration. The automaton is given an initial configuration, after which its configuration repeatedly changes in a sequence of discrete time steps. At each step, all cells are updated simultaneously, according to a pre-specified rule which determines the new value as a function of each cell's previous value and of the values in its two neighboring cells. All cells obey the same rule, which may be given either as a formula or as a rule table that specifies the new value for each possible combination of neighboring values. Each cell new value is the exclusive or of the two neighboring values. Equivalently, the next state of this particular automaton is governed by the following rule table

Figure 6: Rule 90

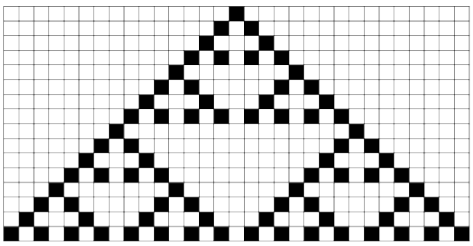

Figure 7: Space Time distribution diagram of Rule 90

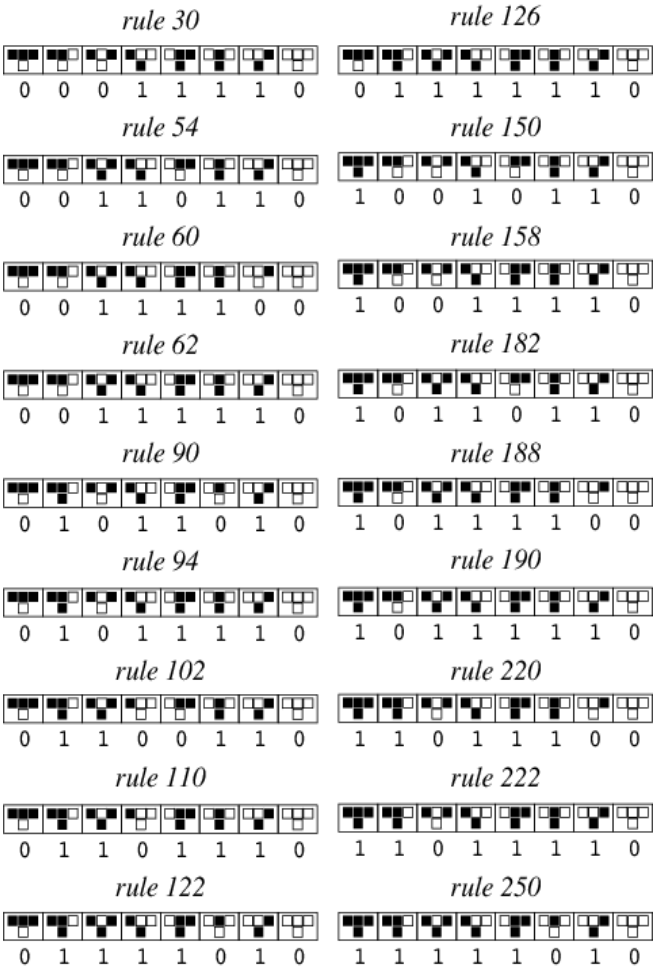

Figure 5 : Rules of Cellular Automata

Rule 90 is an additive cellular automaton: if two initial states are combined by computing the exclusive or of each their states, then their subsequent configurations will be combined in the same way. Thus, more generally, one can partition any configuration into two subsets with disjoint nonzero cells, evolve the two subsets separately, and compute the behavior of the original automaton by superposing configurations derived from the two subsets.

RULE 150

Rule 150 is one of the elementary cellular automaton rules introduced by Stephen Wolfram in 1983 (Wolfram 1983, 2002). It specifies the next color in a cell, depending on its color and its immediate neighbors. Its rule outcomes are encoded in the binary representation. The total activity of the single-seeded cellular rule 150 automaton does not follow a one-step iteration like other cellular automata,

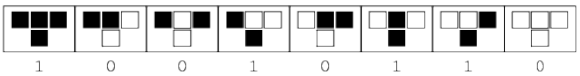

Figure 8: Rule 150

but can be solved as a two-step vectorial, or string, iteration, which can be viewed as a generalization of Fibonacci iteration generating the time series from a sequence of vectors of increasing length. This allows to compute the total activity time series more efficiently than by simulating

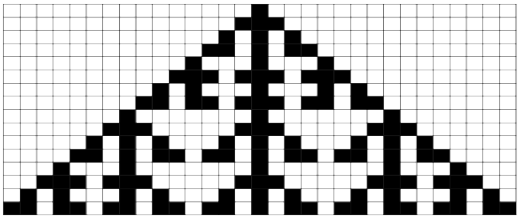

Figure 9 : Space Time distribution diagram of Rule 150

the whole spatio-temporal process, or even by using the closed expression.

The other component of the proposed stream cipher is the key-bit mixing mechanism. The key-bit mechanism as shown in Figure 7, increases the complexity of the stream cipher making it harder to crypto-analyze.

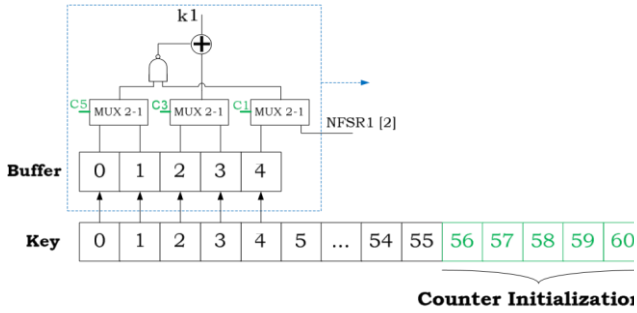

Figure 10: Key bit Mechanism

The output bit K_1 is XORed with the CA bit 4 and provides the feedback to the NFSR. The key is generated using the same process as the A2U2 stream cipher. As shown in the Figure 7, at every round, five bits of the key are loaded into a buffer. Finally, these bits are combined with three bits of the counter and one bit of the NFSR. The last component of the proposed stream cipher is the Filter function.

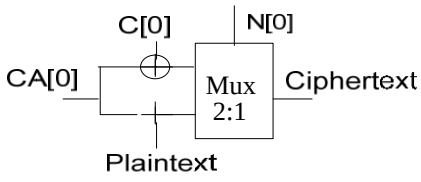

Figure 11: Filter function

The filter function ensures that only the part of the input string coming out from the CA-based register will be XORed with the plaintext based on a selector string provided by the NFSR.

IV RESULTS AND DISCUSSION

The stream cipher is designed using XILINX tool for secure communication in the presence of third parties. The ciphertext generated in this design is very efficient such that it is hard for cryptanalysis. The ciphertext is generated via cellular automata for rules 90 and 150. This is one of the simplest rules in cellular automata for the generation of pseudo random numbers.

V CONCLUSION ANF FUTURE WORK

The performance of a CA-based stream cipher design suitable for implementation on FPGAs. The design was synthesized and tested using a Xilinx Spartan 3E FPGA.

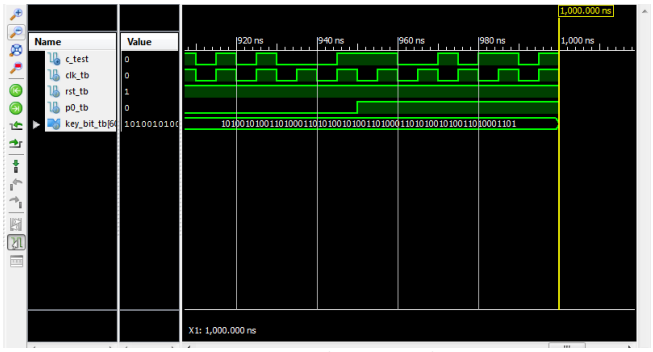

Figure 12: Generation of ciphertext for rule 90

Results were compared with the A2U2 design. The proposed PRNG in this work might be useful in other applications requiring high quality random number sequence. Future work includes cryptanalysis to study the strength of the proposed stream cipher, optimization of the CA design to improve the quality of randomness, and exploration of some of the hardware implementation techniques and to hide data in an image using cellular automata

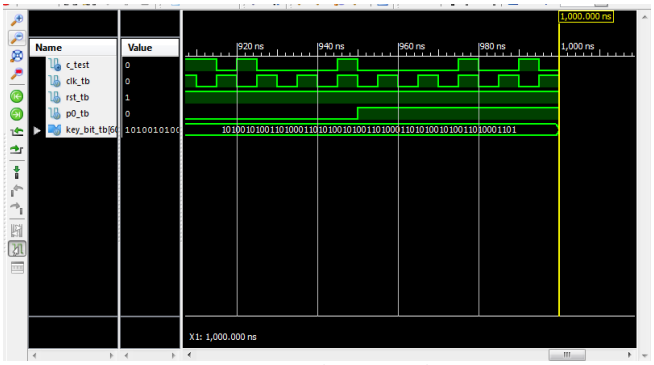

Figure 13: Generation of ciphertext for rule 150

REFERENCES

[1] J. C. Cerda, C. D. Martinez, and J. M. Comer, D. H. K. Hoe, "An Efficient FPGA Random Number Generator using LFSRs and Cellular Automata," IEEE 44th Southeastern Symposium on System Theory, March 2012.

[2] D. Mathieu, D. Ranasinghe, and T. Larsen, "A2U2: A Stream Cipher for Printed Electronics RFID Tags," IEEE International Conference on RFID, 2011.

[3] C. de Canniere, O. Dunkelman, and M. Knezevic , "KATAN & KTANTAN – A Family of Small and Efficient Hardware Oriented Block Ciphers," in Proc. 11th Int. Workshop on Cryptographic Hardware and Embedded Systems-CHES 2009, Switzerland, LNCS, vol. 5747, pp. 272-288.

[4] N. T. Courtois and W. Meier, "Algebraic Attacks on Stream Ciphers with Linear Feedback," in Proc. Workshop Theory and Application of Cryptographic Techniques, Advances in Cryptology – EUROCRYPT '03, Warsaw, Poland, May 4-8, 2003, LNCS, vol. 2656, pp. 345-359, 2003.

[5] R. W. Duren, R. J. Marks II, P. D. Reynolds, and M. L. Trumbo, "Real-Time Neural Network Inversion on the SRC- 6e Reconfigurable Computer," IEEE Trans. on Neural Networks, vol. 18, no. 3, May 2007, pp. 889-901
[6] P. D. Hortensius et al., "Cellular automata-based pseudorandom number generators for built-in self-test," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 8, no. 8, pp. 842-859, Aug. 1989.
[7] S. Wolfram, A New Kind of Science, Wolfram Media, Inc., IL, USA, 2002.