FPGA IMPLEMENTATION OF CRYPTOGRAPHIC SYSTEM USING BODMAS SEQUENCE OF OPERATIONS

Narendra Babu T., Fazal Noorbasha, Sai Krishna Ch., Sai Charan K. and R. S. V. S. Sai Kalyan
K L University Educational Foundation, Vaddeswaram, Guntur, A.P, India
E-Mail: tatininarendra@kluniversity.in

ABSTRACT

In this paper, a new Encryption algorithm is proposed which is high secured and data is transmitted securely over unreliable communication channel. In this algorithm we generate random numbers by using Linear Feedback Shift Register (LFSR) and input data is encrypted by performing sequence of BODMAS operation with random numbers generated by LFSR. The proposed work is simulated, verified and synthesized using Xilinx by Spartan 3E FPGA.

Keywords: BODAMS, cryptographic, operation, encryption algorithm

1. INTRODUCTION

In the existing system the data is first encoded using the orthogonal code for error detection and correction. An LFSR is designed which generate sequence of random numbers. Encoded data is encrypted by performing XOR operation with input data and random number generated by LFSR. Since it performs only one operation, it has low security and can be easily hacked through trial and error method.

So we proposed an encryption algorithm where we perform different arithmetic operation in a sequence according to BODMAS rule with the random numbers generated by the LFSR. This improves the level of security.

2. RELATED WORK

Narendra Babu T proposed a new algorithm for error detection and correction rate of the data. In this algorithm the input is encoded to orthogonal code and then encrypted using LFSR [1].

Saikat Chakrabarti presented an efficient signature scheme called CLFSR-QA, where it offers a constant length signature with least storage elements [2].

Enes Pasalic proposed a technique where we apply a guess and determine the attack with LFSR based ciphers are analyzed, this technique is based on guessing certain bits in order to determine the remaining secret bits [3].

François Arnault proposed a technique which uses linear finite state machines (LFSM) for cryptography. These LFSM are represented in matrices and those are represented using polynomial. This new algorithm picks random numbers from LFSR with good properties [4].

Rimba Whidiana Ciptasari proposed a technique for the audio owner protection by using the visual cryptography. In this technique they use public and secret images for encrypting the audio data with the help of discrete wavelet transform and discrete cosine transform [5].

Subhas Barman proposed a new encryption technique which is based on fingerprint. In this approach the sender and receiver shares their fingerprint securely and they are convoluted and it is used as key for encryption [6].

Munetoshi Iwakiri proposed a technique for copy right protection where fragile watermarking based on incomplete cryptography. Fragile watermarking is included in the encryption and decryption of digital content [7].

Massoud Masoumi discussed about the contribution of the nano technology offers to the evolution of cryptography and embedded systems and demonstrates about the nano devices are used in constructing the sensitive devices [8].

Harsha S. Jois proposed a cryptographic system which is independent to the file type. This system uses Cartesian and polar coordinates in three dimensions. This can be effectively used for medical images, encryption based on geometrical objects [9].

Samta Gajbhiye discussed about the Elliptic curve Cryptography (ECC) and various forms of ECC in various coordinate system [10].

Safwat Hamad proposed an encryption technique where the text is transformed into single standard DNA sequence and then encrypted using the 8x8 codon matrix. An interweaving step is added which improves the randomness [11].

Akshey Nanda proposed a methodology for cryptography which mainly aims to protect the data from phishing. In this technique the data is authenticated using the fingerprint so that the phishing the data is avoided [12].

Yasushi Yamaguchi proposed a scheme for a extended visual cryptography where the data can be decrypted only by human visual system. The scheme based on encryption of images with no pixel expansion [13].

Hari Narayan Khan proposed a new encryption model, where session keys are used to encrypt the data. For every session new key is used for encryption. The key is based on function of secret key and current plain text [14].

Gaurav Indra proposed a cryptosystem which is based on mutual authentication and key management. Elliptical curve cryptography is used with a novel time stamp mechanism for the encryption of the data [15].

3. PROPOSED WORK

The proposed Encryption technique is based on the BODMAS rule. We perform sequence of arithmetic operations.

- Brackets (calculation inside brackets always comes first).
- **Or**ders (powers or square roots).
- Division.
- Multiplication.
- Addition.
- Subtraction.

In this encryption we are considering operations Division, Multiplication, Addition and Subtraction. Let us assume m-bit LFSR. The LFSR is designed using the Exclusive OR operation (XOR) therefore the zero is excluded. No of random numbers generated are 2^m-1

The design of LFSR depends on the no of operations we are going to perform. Let us assume 'N' no of operations.so the LFSR designed is according to the 'N'.

2^m-1 >= N

Let the tapping connections of LFSR is LSB and MSB. Therefore the polynomial of LFSR is

X^m + X + 1

Let the input data (X) a-bit data is divided with first random number (R1), we get quotient and remainder.

X / R1 (X – a bit, R1 – m bit data)

The maximum size of quotient and remainder will also be a-bit. Now both quotient and remainder are concatenated to give 2* a-bit data. This is stored in a variable DIV. Let this be called b-bit data (2 * a-bit).

DIV = {quotient (a bit), remainder (a bit)}

This data (DIV) is multiplied with second random number (R2) and the result is stored in another variable MUL.

MUL = DIV * R2

Maximum size of this is 2^b *2^m. Size of MUL is **(b + m)** bit data.

This data (MUL) is added with the third random number (R3) and the result is stored in a variable ADD.

ADD = MUL + R3 (Max. Size is b+m+1)

This ADD is subtracted by the fourth random number (R4) and the final result is stored in a variable CIPHER.

CIPHER = ADD – R4 (Max. Size is b+m+1).

This CIPHER is transmitted over the communication channel to receiver. Receiver decrypts the encrypted data by performing the operations opposite to that of transmitter. Sequence of operations to be performed on receiver side is Addition with R4, Subtraction with R3, Division with R2 and Multiplication with R1.

4. ALGORITHM

4.1 At Transmitter

Let the input is a 4 bit data and LFSR is a 3 bit LFSR designed using XOR. Therefore 7 random numbers are generated (R1 to R7), zero (000) is excluded from the sequence of random numbers.

The polynomial of designed LFSR is $\boldsymbol{X^3}$ + X + 1

Step 1:

Input 4 bit data is stored in a variable PLAIN_TEXT. The input is divided with the first random number (R1). We get two values quotient and remainder, and these are stored in variables QUOTIENT and REMAINDER.

PLAIN_TEXT / R1

Step 2:

Both the QUOTIENT and REMAINDER are concatenated to a single variable DIV.

DIV = {QUOTIENT, REMAINDER}

Step 3:

The DIV is multiplied with the 2nd random number (R2) and the result is stored in MUL variable.

MUL = DIV * R2

Step 4:

MUL is added with the 3rd random number (R3) and the result is stored in ADD variable.

ADD = MUL + R3

Step 5:

Finally the ADD is subtracted by the 4th random number (R4) and stored in variable CIPHER_TEXT.

CIPHER_TEXT = ADD – R4

4.2 At Receiver

The LFSR is designed with same tapping connections to get the same sequence of random numbers (R1 to R7). The CIPHER_TEXT is received from the transmitter over the communication channel.

Step 1:

This CIPHER_TEXT is added with the 4th random number (R4) and the result is stored in the ADD variable.

ADD = CIPHER_TEXT + R4

Step 2:

ADD is now subtracted by the 3rd random number (R3) and the result is stored in a variable SUB.

SUB = ADD – R3

Step 3:

SUB is divided by the 2nd random number (R2) and gives quotient and remainder, which are stored in QUOTIENT and REMAINDER.

QUOTIENT = SUB / R2

REMAINDER = SUB % R2

Step 4:

Finally the REMAINDER is multiplied with the 1st random number (R1) and then QUOTIENT is added to it to get back the original data. This is stored in PLAIN_TEXT.

PLAIN_TEXT = ((REMAINDER * R1) + QUOTIENT)

These are the steps to decrypt the encrypted data.

5. IMPLEMENTATION

Let the input data be PLAIN_TEXT = 0011. Random numbers generated by LFSR are 001(1), 100(4), 110(6), 111(7), 011(3), 101(5), and 010 (2)

Step 1:
PLAIN_TEXT is divided by R1
QUOTIENT = 0011 / 001
= 0011
REMAINDER = 0011 % 001
= 0000

Step 2:
Concatenating REMAINDER and QUOTIENT are stored in DIV
DIV = {0000, 0011} = 00000011

Step 3:
Multiplying DIV with R2
MUL = 00000011 * 10
= 00000001100

Step 4:
Addition of MUL with R3
ADD = 00001100 + 110
= 000000010010

Step 5:
Subtracting ADD by R4
CIPHER_TEXT = 00010010 – 111
= 000000001011

The simulation output of the transmitter is shown in the Figure-1. This is transmitted to receiver at other end. Receiver decrypts the data to get back the original data.

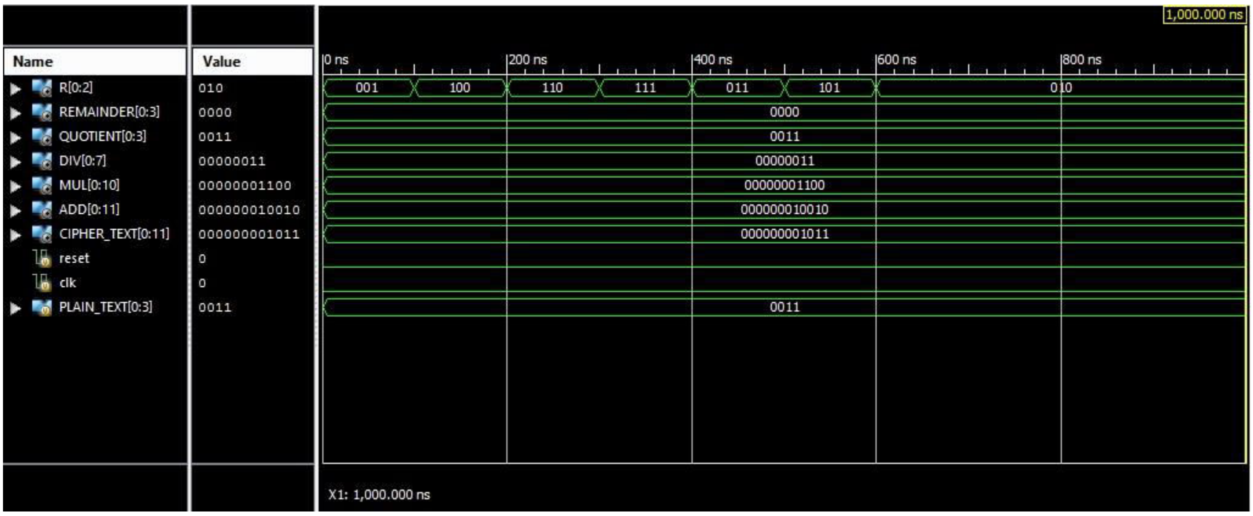
Step 1:
Addition of CIPHER_TEXT with R4
ADD = 00001011 + 111
= 000000010010

Step 2:
Subtracting ADD with R3
SUB = 00010010 – 110
= 000000001100

Step 3:
Division of SUB with R2
QUOTIENT = 00001100 / 100
= 0011
REMAINDER = 00001100 % 100
= 0000

Step 4:
Multiply REMAINDER with R1 and then add QUPTIENT to get the original data
PLAIN_TEXT = ((0000 * 001) + 0011)
= 0011

The simulation output of the receiver is shown in the Figure-2. The cipher data is decrypted successfully and the original data is obtained.



Name	Value
R[0:2]	010
REMAINDER[0:3]	0000
QUOTIENT[0:3]	0011
DIV[0:7]	00000011
MUL[0:10]	00000001100
ADD[0:11]	000000010010
CIPHER_TEXT[0:11]	000000001011
reset	0
clk	0
PLAIN_TEXT[0:3]	0011

Figure-1. Simulation output of the transmitter.

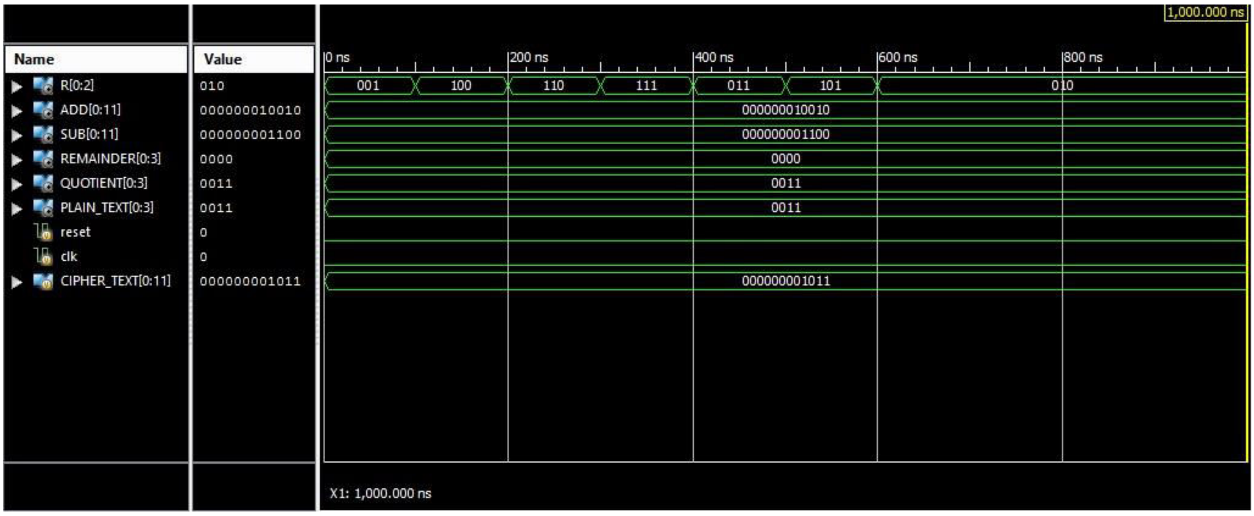


Figure-2. Simulation output of the receiver.

Table-1. Summary of decryption time for various input data length in existing system.

S. No.	**Input data** (m-bit)	**LFSR size** (m-bit)	**M =2^m-1**	**Time taken to decrypt one instruction** (msec)	**Time taken to detect input** (msec)
1	3	3	7	1	7
2	4	4	15	1	15
3	5	5	31	1	31
4	6	6	63	1	63
5	7	7	127	1	127
6	8	8	255	1	255
7	M	M	2^m-1	1	2^m-1

Table-2. Summary of decryption time for various input data length in proposed system.

S. No.	**Input data** (m-bit)	**LFSR size** (m-bit)	**M =2^m-1**	**No of arithmetic operations** (N)	**Time taken to detect input** (msec)
1	3	3	7	4	840
2	4	4	15	4	32760
3	5	5	31	4	755160
4	6	6	63	4	14295960
5	7	7	127	4	248031000
6	8	8	255	4	4129476120
7	M	M	M= 2^m-1	4	M_{C_N}

6. COMPARISONS

Table-1 shows the Summary of Decryption time for various input data length in existing system. In the Existing system, m-bit LFSR generates (2^m-1 = M) random numbers. The input data is XOR'd with the random number. So there is a possibility to hack the data by performing the trial and error. By performing trial and error M times with all random numbers then we can know the input data. Let trial and error of one random number take 1 mille second. So in existing system the input data can be hacked in M mille seconds.

Table-2 shows the Summary of Decryption time for various input data length in proposed system .In Proposed system, LFSR is designed m-bit which gives (2^m-1 = M) random numbers. We perform N arithmetic operations with input data and the random numbers. To hack this system the no of combinations required for trial and error is M_{C_N} . So time take to hack the input data is much higher than the existing system and the security level is improved.

The proposed technique is simulated, verified and synthesized using the Xilinx ISE 14.1 and is implement on the Spartan 3E FPGA. The simulation results of transmitter and receiver are shown in the Figure-1 and Figure-2.

7. CONCLUSIONS

The results of the proposed work show that time taken to detect the input bits increases drastically with respect to the size of LFSR (m-bit). More time it takes to detect the input security is higher. When compared with existing work the security is much higher in the present work.

REFERENCES

[1] Narendra Babu T. 2016. Implementation of High Security Cryptographic System with Improved Error Correction and Detection Rate using FPGA. 6(2): 602~610, IJECE.

[2] Saikat Chakrabarti. 2009. An Efficient and Scalable Quasi-Aggregate Signature Scheme Based on LFSR Sequences. 20(7), IEEE Transactions.

[3] Enes Pasalic. 2009. On Guess and Determine Cryptanalysis of LFSR-Based Stream Ciphers. 55(7), IEEE Transactions.

[4] François Arnault. 2011. Revisiting LFSRs for Cryptographic Applications. 57(12), IEEE Transactions.

[5] Rimba Whidiana Ciptasari. 2014. An enhanced audio ownership protection scheme based on visual cryptography. EURASIP Journal on Information Security. 2014: 2.

[6] Subhas Barman. 2015. Fingerprint-based crypto-biometric system for network security. EURASIP Journal on Information Security. 2015: 3.

[7] Munetoshi Iwakiri. 2015. Fragile watermarking based on incomplete cryptography for copyright protection. Applied Informatics. 2: 7 DOI 10.1186/s40535-015-0012-8.

[8] Massoud Masoumi. 2015. Nanoscale cryptography: opportunities and challenges. Nano Convergence. 2: 21.

[9] Harsha S. Jois. 2015. A 3-d advancement of PythoCrypt for any file type. Journal of Open Innovation: Technology, Market, and Complexity. 1: 19.

[10] Samta Gajbhiye. 2011. A Survey Report on Elliptic Curve Cryptography. 1(2): 195~201, IJECE.

[11] Safwat Hamad. 2014. A Novel Implementation of an Extended 8x8 Playfair Cipher Using Interweaving on DNA-encoded Data. 4(1): 93~100, IJECE.

[12] Akshey Nanda. 2015. Anti-Phishing Techniques in Cryptography. 5(6): 1511-1515, IJECE.

[13] Yasushi Yamaguchi. 2015. Extended visual cryptography for continuous-tone images: effect of the optimum tone mapping. Int. J. Information and Communication Technology. 7(1).

[14] Hari Narayan Khan. 2015. Robust symmetric cryptography using plain-text variant session key. Int. J. Electronic Security and Digital Forensics. 7(1).

[15] Gaurav Indra. 2014. A time stamp-based elliptic curve cryptosystem for wireless ad-hoc sensor networks. Int. J. Space-Based and Situated Computing. 4(1).