ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 7, Issue 12, December 2020

Implementation of Encryption and Decryption using Random Chosen Polynomial

[1] Prema B, [2] Sastry K.R.K

[1] M.TECH, Electronics and Communication Engineering, JNTUK University, Gayatri Vidya Parishad college of engineering (Autonomous), Madhurawada, Visakhapatnam, Andhra Pradesh, India

[2] Associate professor, Electronics and communication engineering, JNTUK University, Gayatri Vidya Parishad college of engineering (Autonomous), Madhurawada, Visakhapatnam, Andhra Pradesh, India

Abstract: In recent years, there has been a growing trend for software applications to use representative service providers for content distributions. These representative services have brought new tasks for the safety of software content concealment. Encryption is a technique which can be used to conceal information from unofficial individuals, either in storage or in communication. This technique transforms the content into a jumbled or an un-viewable format. The proposed scheme is based on randomly chosen polynomial. The new algorithm offers encryption and decryption with sufficient security and so data is at low computing overhead.

Index Terms—Encryption, Decryption, random chosen polynomial, Cipher

INTRODUCTION

Cryptography is the method to secure information from third parties called adversaries. Most common, cryptography is about creating and evaluating protocols that prevent adversaries or the public from reading confidential messages. Various features in data safety such as data concealment, data integrity, validation is a center to modern cryptography. Modern cryptography exists at the connection of the disciplines of mathematics, computer science, electrical engineering, communication engineering, and physics. In present years, there had been a growing tendency for software applications to use representative service providers for contented distributions. These representative services had brought new tasks to the safety of software content confidentiality.

The proposed scheme is based on randomly chosen polynomial. The new algorithm offers a onetime encryption with sufficient security. The proposed scheme is based on polynomial generation. The polynomial used is generated randomly and is of degree n, where n is a positive integer. Cryptography is a method of safeguarding data and communications through practice of codes so that only those people for whom the data is planned can understand it and process it. Hence avoiding illegal access to information. The word “crypt” defines “hidden” and graphy defines “writing”. In Cryptography, the methodologies which are used to secure data are achieved from mathematical perceptions and a set of rule-based calculations known as algorithms to change messages in such a way that makes it hard to decode it.

Random Chosen Polynomial

The polynomial equation is chosen by selecting the coefficients. By using the randomly chosen polynomial encryption and decryption is done. Encryption and decryption are done by using private key. Encryption is done by changing plain text to encryption text and decryption is done by converting encryption text to plain text.

Functionality

Message is sent to the sender. The sender using the encryption algorithm encrypts the message using a private key. Then, the plain text is converted to the cipher text. Now, using the private key the cipher text is received. The ciphered text is sent to the receiver, the receiver using the decryption algorithm decrypts the ciphered text to the plain text. The plain text is achieved using the private key. The message is sent in the form of blocks.

Overall block diagram

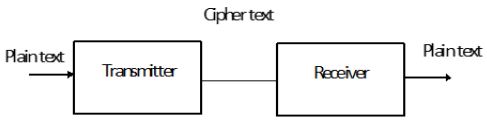


fig.1 proposed block diagram

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 7, Issue 12, December 2020

Plain text is given as input to the transmitter. The transmitter performs encryption algorithm and the output of the transmitter is cipher text. The cipher text is given as input to the receiver, therefore the receiver performs decryption algorithm. Hence the output of the receiver is plain text.

IMPLEMENTATION OF ENCRYPTION

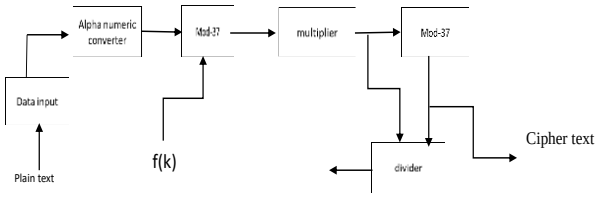


Fig.2 Block diagram for Transmitter

Plain text data is given as input to the data input block (ak). The output is given to the alpha-numeric block converter. The data is sent in the form of blocks. The output of the data input block is given as input to the alpha-numeric converter block. The alpha-numeric block maps the each block of data to alpha-numerical values. Basically, the alpha-numeric values are assigned to numerical. The following characters are assigned to numerical.

A=0,B=1,C=2,----,Z=25,0 = 26, 1 = 27, ……., 9 = 35 and space = 36.

We have chosen mod 37 because the total count of alpha-numerical are 37. Two inputs are taken to perform the modulus operation. f(k) is the polynomial equation with coefficients, Where k=1,2……..

Generally, multiplication operation requires two inputs. Therefore, the output of modulus block is given as one of the input and the output of alpha-numeric converter block is given as another input to the multiplier. Finally, the multiplication output is "b_k". Two inputs are taken to perform the modulus operation. One input 37 and another input is the output of the multiplier. The obtained output is the cipher text (d_k). Division operation is performed in order to know the private key. The output the multiplier block and the modulus 37 block is given as the two inputs to the divider. Thus obtained private key(l_k) is the output of the divider block.

Procedural steps for encryption

1. Chosen a prime number as our modulus.
2. Mapped the alphabets, digits and numbers to alpha numeric values.
3. Randomly chosen the polynomial coefficients.
4. As a result of chosen coefficients, polynomial is known.
5. Let M be the plain text message of width 'm', so that the message can be written as M: a1, a2, a3, ---, a_m , where a_i is the i^{th} letter in the message M. Here ai may be an alphabet or a digits or a space.
6. We convert the plain text M to alpha-numerical values .
7. Let the converted text 'C' be given by C: c1, c2, c3, ---, cm. where ci is the converted value of ai.
8. Now we calculate G: g1, g2, g3, ---, gm by using f (k) = g_k mod 37 where k = 1, 2, 3, ----, m.
9. Then g_k lies between 0 and 36 (0< 37).
10. Here D: d1, d2, d3, ---, d_m is the ciphered message numerical values or encrypted text message values.
11. Here l1, l2, l3, ---, l_m, is the private key.
12. We now convert the cipher text message values 'D' back to alpha numeric string.
13. Thus we get cipher text message.

IMPLEMENTATION OF DECRYPTION

The output of the data input block is given as input to the alpha-numeric converter block. The alpha-numeric block maps the each block of data to alpha-numerical values. Basically, the alpha-numeric values are assigned to numerical. The following characters are assigned to numerical.

A=0,B=1,C=2,----,Z-25,0 = 26, 1 = 27, ……., 9 = 35 and space = 36. The polynomial equation is generated and the coefficients are chosen randomly. The polynomial equation is defined as f(k) where k=1,2,…..n. We have chosen mod 37 because the total count of alpha-numerical are 37.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 7, Issue 12, December 2020

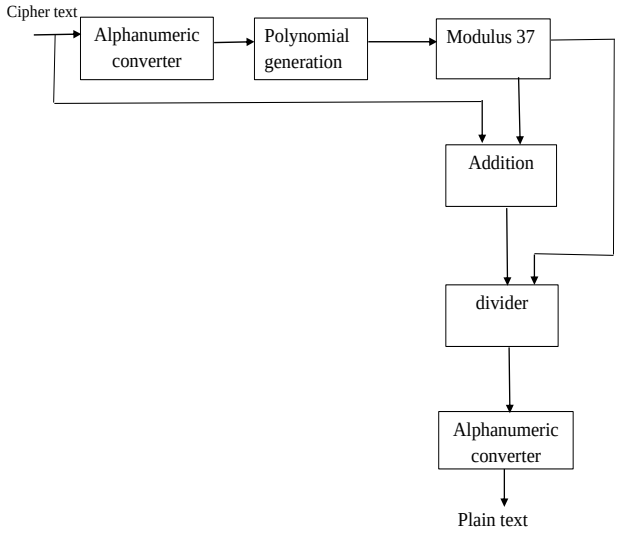


Fig.3 Block diagram for Transmitter

In order to perform addition, two inputs are taken. The output of the modulus 37 is given as one of the input and the other input is taken from the input of the alpha-numeric block. Division operation is performed in order to know the private key. The output the multiplier block and the modulus 37 block is given as the two inputs to the divider. Thus obtained private key(lk) is the output of the divider block. The output of the data input block is given as input to the alpha-numeric converter block. The alpha-numeric block maps the each block of data to alpha-numerical values.

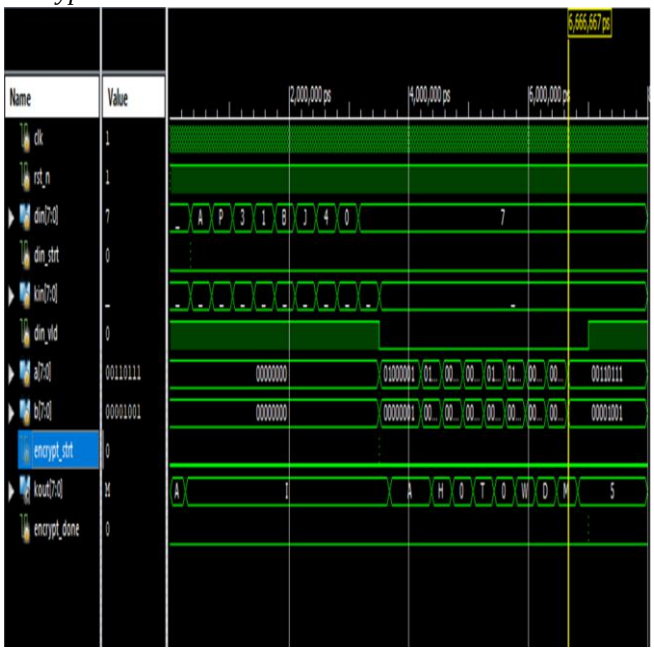
PROCEDURAL STEPS FOR ENCRYPTION

1) The receiver has received the cipher text and private key.
2) Convert the cipher text to alpha-numerical string.
3) Calculate f(k) by using private key and corresponding polynomial coefficients p0, p1, p2, ---, p_n, where k = 1 to 10.
4) Compute g_k via f (k) • g_k mod 37. we get g_k values lying between 0 to 36.
5) Since D: d1, d2, d3, ---, d_m is known.
6) Now we calculate b k = 37l k + d k using private key L: l1, l2, l3, ---, l_m.
7) Evaluate ck.
8) Note that g_k • 0 because when g_k = 0, we should replace g_k by 37. where ck is decrypted message numerical values.
9) Covert the decrypted message to alpha - numerical string.

RESULTS AND ANALYSIS

The design is technologically advanced in Verilog HDL (Hardware Description Language), synthesized and simulated by using Xilinx 14.3 software.

Encryption Simulation



fig

Fig 4. Encryption output

Decryption simulation

ISSN (Online) 2394-2320

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)

Vol 7, Issue 12, December 2020

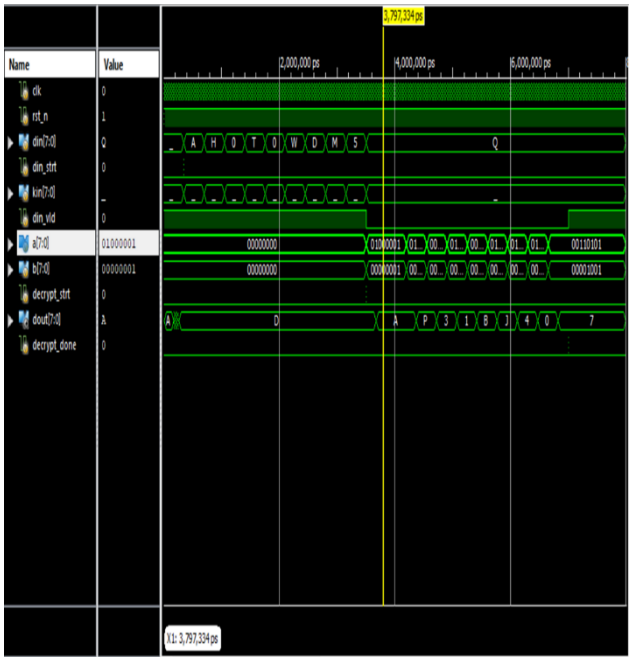


Fig 5. Decryption output

Timing Summary of Encryption

1. Speed grade: -4
2. Minimum period: 10.380ns
3. Minimum input arrival time before clock: No path found
4. Maximum output required time after clock: 4.283ns
5. Maximum combinational path delay: No path found

Device utilization summary of Encryption

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1121	4656	24%
Number of Slice Flip Flops	1821	9312	19%
Number of 4 input LUTs	804	9312	8%
Number of bonded IOBs	18	190	9%
Number of GCLKs	1	24	4%

Timing Summary of Decryption

1. Speed grade: -4
2. Minimum period: 9.594ns (Maximum Frequency: 104.232MHz)
3. Minimum input arrival time before clock: No path found
4. Maximum output required time after clock: 4.283ns
5. Maximum combinational path delay: No path found

Device utilization summary of Decryption

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	931	9,312	9%
Number of 4 input LUTs	497	9,312	5%
Number of occupied Slices	678	4,656	14%
Number of Slices containing only related logic	678	678	100%
Number of Slices containing unrelated logic	0	678	0%
Total Number of 4 input LUTs	518	9,312	5%
Number used as logic	458		
Number used as a route-thru	21		
Number used as Shift registers	39		
Number of bonded IOBs	10	190	5%
IOB Flip Flops	1		
Number of BUFGMUXs	1	24	4%
Average Fanout of Non-Clock Nets	1.54		

CONCLUSION

In this Encryption and Decryption is carried out using an random chosen polynomial. we had used block cipher, in order to send the data in the form of blocks. Different patterns of data is sent to the transmitter, and the output we get from the transmitter is cipher text.

The received output from transmitter is given as the input to the receiver and the output of the receiver is cipher text. The polynomial used here is chosen randomly by taking the respective coefficients.

REFERENCES

1) "A Cryptographic Scheme of Laplace Transforms", G. Naga Lakshmi, B. Ravi Kumar and A. Chandra Sekhar, March 2015, IJRPC.

2) Implementation of elliptic curve cryptography in 'c', IJET, Kuldeep bhardwaj and Sanjay Chaudhary Department of Mathematics, Dr. B. R. Ambedkar University, IBS, Khandari, Agra (UP).

3) Implementation of Text Encryption using Elliptic Curve Cryptography, Laiphrakpam Dolendro Singh and Khumanthem Manglem Singh, National Institute of Technology, Manipur, Imphal East 795 001,

India.

4) Elliptic Curve Cryptography for Secured Text Encryption, Keerthi K, Department of CSE, National Institute of Technology, Puducherry India, Dr. B. Surendiran, Department of CSE, National Institute of Technology, Puducherry India.

5) Fast Elliptic Curve Cryptography on FPGA, William N. Chelton, Student Member, IEEE, and Mohammed Benaissa, Senior Member, IEEE.

6) FPGA Based Implementation of Elliptic Curve Cryptography, Mustafa Nawari, Hazim Ahmed, Aisha Hamid, Mohamed Elkhidir, Department of Electrical & Electronic Engineering, University of Khartoum, Khartoum, Sudan.

7) High Performance FPGA Implementation of Elliptic Curve Cryptography over Binary Fields, Shuai Liu, Lei Ju, Xiaojun Cai, Zhiping Jia, Zhiyong Zhang, School of Computer Science and Technology, Shandong University, Jinan, China.

8) FPGA-Based Efficient Modular Multiplication for Elliptic Curve Cryptography, Md Selim Hossain and Yinan Kong, Department of Engineering, Macquarie University, Sydney, Australia.