International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 2, April 2016, pp. 602~610
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i2.9267

Implementation of High Security Cryptographic System with Improved Error Correction and Detection Rate using FPGA

Narendra Babu T*, Fazal Noorbasha*, Leenendra Chowdary Gunnam**
* Department of Electronics and Communication Engineering, K L University, Guntur, Andhra Pradesh, India
** Department of Electronics and Communication Engineering, Sasi Institute of Technology and Engineering, India

Article Info

Article history:

Received Oct 27, 2015
Revised Dec 14, 2015
Accepted Jan 8, 2016

Keyword:

Encryption
FPGA
LFSR
Orthogonal code

ABSTRACT

In this article, an encryption algorithm with an error detection technique is presented for highly secured reliable data transmission over unreliable communication channels. In this algorithm, an input data is mapped into orthogonal code first. After that the code is encrypted with the help of Linear Feedback Shift Register (LFSR). The technique has been successfully verified and synthesized using Xilinx by Spartan-3E FPGA. The results show that the error detection rate has been increased to 100% by proposed encryption scheme is effective and improves bandwidth efficiency.

Corresponding Author:

Narendra Babu T,
Departement of Electronics and Communication Engineering,
K L University, Guntur, Andhra Pradesh, India.

1. INTRODUCTION

1.1. Cryptography

Cryptography is a process of transmitting stored data in a particular form i.e. ciphers text where only the permitted person can access it. Simply we can say that it is the process of securing data by scrambling into an incomprehensible arrangement, called cipher text (encryption). Just the individuals who have a secret key can decode the message into plain content (decryption). Encryption can be defined as changing the original message into other form which can be again retained by using the key. Decryption is the process of converting cipher text back to plaintext. The originator of a encrypted message imparted the decoding procedure expected to recoup the first data just with proposed beneficiaries, accordingly blocking undesirable persons todo similar. Encoded messages can here and there be broken by cryptanalysis, likewise called code breaking. Cryptographers are the name given to those who practice this type of cryptographic system.

Cryptography concerns mainly with four characteristics they are confidentiality, integrity, non-repudiation, and authentication. Only those systems and protocols which satisfy all the above mentioned characteristics are known as cryptosystems. Cryptosystems are often thought to refer only to mathematical procedures and computer programs. Cryptosystems are classified into two categories they are symmetric key and asymmetric key or public key cryptographic systems.

1.2. Error Detection Codes

Error detection and correction are the methods that enhance the secured delivery of digital data over unsecured communication channels. In the field of information theory and coding where the applications of computer science and telecommunications play an important role in using these error detection techniques. During transmission most of the communication channels add on to some additional data to the original data

which is predominantly known as noise. Due to this many unwanted data or errors occur during the transmission of data from sender to receiver. In order to limit this problem we have error detection and correction techniques. The error detecting method is used to detect that particular error while the error correction methods are used to reconstruct the original data. Generally the definition of these terms is given below.

Error detection: is defined as the method of detecting the errors which are caused due to external noise or any other sources.

Error correction: is defined as the method of reconstructing the original data and transmits the error free data to the receiver.

Some of the existing EDC codes are Hamming code and Inverse Gray Code are single bit error correcting code. Advanced hamming code is single bit error correcting code and double bit error detection. Residue number system is a single bit error correcting code used in DSP processors.

Proposed EDC code is orthogonal code. These codes are binary valued codes which has equal number of 0's and 1's.therefore all orthogonal code generates zero parity bits. If there is transmission error, then we can find the error by generating the parity bits at the receiver end. The parity generation method can detect only 50% ($\frac{2^n}{2}$) errors. Since parity bit does not effect for the even number of errors. My approach is to compare the incoming orthogonal code with the all the orthogonal codes which are stored in the look up table at the receiver end for the possible match.

This correlation process is given by

$$R(x, y) = \sum_i^b x_i y_i \leq \frac{b}{4} - 1$$

The average number of errors that can be corrected is given by

$$t = b - R(x, y) = \frac{b}{4} - 1$$

The below table shows the error correction capabilities.

Table 1. Shows the error correction capabilities

S.no	a-bit data	b = 2^(a-1)	t = (b/4)-1
1	4-bit data	8-bit code	1
2	5-bit data	16-bit code	3
3	6-bit data	32-bit code	7
4	7-bit data	64-bit code	15
5	8-bit data	128-bit code	31

1.3. Encryption Technics

The existing Encryption technics are Data Encryption Standard(DES) where it uses a 64 bit private key, Advanced encryption standard where it uses variable length 128, 196, 256 bit keys, DES40 where key is pre-processed to provide 40 bit key, MD5 where it is used to encrypt one time passwords, it uses 128 bit key. In the proposed encryption technique, data encryption is based on random number which is generated by the Linear Feedback Shift Register (LFSR). LFSR is a shift register whose input is a linear function of its previous state. Therefore the feedback polynomial decides the pseudorandom pattern of the LFSR.

The characteristic polynomial of the LFSR is

$$g(x) = g_r x^r + g_{r-1} x^{r-1} + \text{----} + g_2 x^2 + g_1 x + g_0$$

The remainder of this paper is organized as follows: In section 2 we have reviewed the references on the error detection and correction and on various cryptographic techniques. Section 3 presents the proposed design methodology for error detection/correction and encryption using LFSR. Section 4 presents the implementation and results of the proposed design.

2. RELATED WORK

Pedro Reviriego *et al* proposed an alternative scheme to achieve single error correction (SEC) code in memories. The approach is based on the use of parity sharing which is also formed by a combination of two codes. In this case, the codes use a parity bit and a SEC code [1].

Rukmani R and M Jagadeeswari proposed a design for Error detection and correction architecture to detect the multiple errors and recover the data in motion estimation. It is the process of describing the motion vectors in the transformation of one 2D image to other. The design is based on the residue-and-quotient [2].

Jayarani M.A and Jagadeeswari M proposed a design of Majority logic detector/decoder for faulty detection along with correction of memory applications. This is done by one step logic decoding [3].

Constantin Anton and his team present a solution for the errors correction using regular LDPC and Hopfield network based associative memories. Their solution solves this problem by using an associative memory based on the Hopfield network on the decoding stage, which stores the correct code words. This memory tends to transform the code words received with errors in errors free code words [4].

Luis-J Saiz-Adalid proposed the use of Hamming codes modified to detect 2-bit and 3-bit burst errors, maintaining the single error correction (SEC) feature, with no extra redundancy and having the same encoder and decoder latencies. The codes proposed add the ability to detect short burst errors, maintaining the same redundancy and latency, and with a slight increase in the complexity of the decoder circuit [5].

S.Baskar, proposed a scheme for fault-detection and correction method significantly makes area overhead minimal and to reduce the decoding time through DC codes than the existing technique and it gives promising option for memory applications [6].

Bo Dai and Zhensen Gao proposed a communication system based on the orthogonal differential phase shift keying with public key cryptography. The privacy of transmitted data is doubled guaranteed by this cryptography [7].

Lamonica M proposed a cryptography technique where we use quirky laws of quantum physics to encrypt the data. This cryptography can secure point-to-point connections only about 100 km [8].

Sukalyan som and Sayanisen proposed a Non-adaptive Partial Encryption of Gray scale Images Based on Chaos. In this technique the gray scale images are decomposed into binary 8 bit planes and the decrypted using couple tentmap based pseudorandom binary number generator [9].

Hossein Rahmani proposed the technique used is to create, XaaS concept, we design an Encryption as a Service in order to get rid of the security risks of cloud provider's encryption and the inefficiency of client-side encryption [10].

Marius Iulian and Mihailescu came out with the technique used is to create a strong and unique authentication process of the biometric templates and to guarantee the safety of the biometric data [11].

Xiaotian Wu, Wei Sun proposed a xor based visual cryptography (VC) which is used to solve the poor visual quality problem, two XOR based VC's are proposed, XOR based VC for general access structure and adaptive region incrementing XOR based VC [12].

Syed Rizvi, Katie Cover proposed a Encryption technique especially for cloud service providers where the encryption scheme combines both symmetric and asymmetric cryptographic algorithms which provides strong data confidentiality preserving secret key encryption functionalities [13].

Xuanxia Yao and Zhi Chen proposed the technique used is a lightweight no-pairing ABE scheme based on elliptic curve cryptography (ECC). This scheme proposed to address the security and privacy issues in IoT [14].

Gilles Brassard proposed the technique used is to create a radically different foundation for cryptography and the uncertainty principle of quantum physics. In conventional information theory and cryptography [15].

Zhang Qiming proposed the technique presents a digital certificate which is an electronic document that provides security services using public key, this key is used for encryption or authentication of signature to the private key. This paper uses a tool called C language generation public key algorithm for the authentication of the private key [16].

P.K. Das proposed correcting the errors in codes which are to be transmitted from source to destination using parity bits. Adding the parity bits to the existing codes which is in different orders using theorems, which gives a bound on the requirement number of parity bits [17].

3. RESEARCH METHOD

Since LFSR generates pseudo random numbers where each number is used to encrypt the a-bit input data. Our approach is not only to encrypt the input data but also to encode the data before the encoding into orthogonal code. Once the input data is encoded, the encoded data is XOR'ed with the random number generated by LFSR method. This following approach is to improve the error detection rate by increasing its reliability. Orthogonal code technique involves major blocks i.e. transmitter and receiver which are described below.

Transmitter:

An encoder, encryptor and a shift register the entire three blocks combine to form the basic transmitter in this method. The output of encoder is set to $b=2^{a-1}$ bits where the input is a-bit data which is fed to the encoder. This b-bit data is sent to the encryptor where it is XOR'ed with the pseudo random number generated by the LFSR. In order to transmit this code need to be changed as a serial. For this transformation we use the shift register as shown in the figure 1. Thus by using raising edge of the clock pulse the generated orthogonal code is transmitted using the shift register.

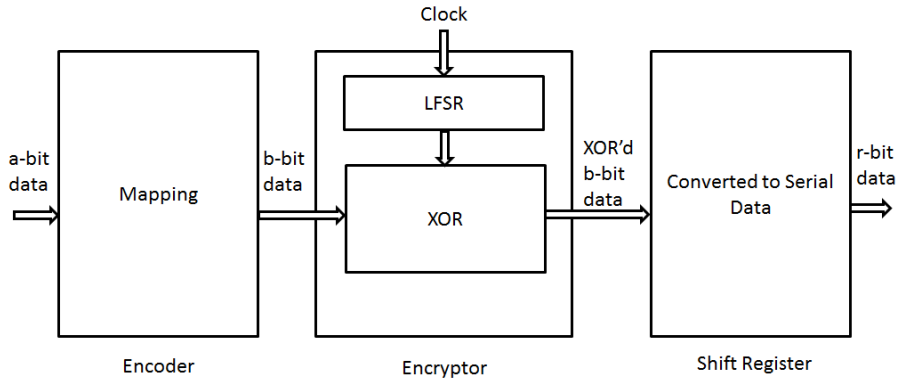

Figure 1. Block diagram of transmitter

Receiver:

The inverse arrangement of the transmitter acts as a receiver i.e. a shift register, decryptor and decoder combine to form the basic receiver component. The received code is processed through the following sequential steps as shown in figure 2. The incoming serial bit data is converted into r-bit parallel codes by using a shift register. The r-bit data is given to the decryptor. In the decryptor the r-bit data is XOR'ed with the random number generated by the LFSR in synchronous with clock and generates an output of b-bit data. The b-bit data is given to the decoder where it is compared with all the codes stored in the lookup table for error detection. This is done by counting the number of ones in the signal resulting from 'XOR' operation between the b-bit data and each combination of the orthogonal codes in the lookup table. A counter is used to count the number of ones in the resulting data and searches for the minimum count. However a value rather than zero shows an error in the received code. The orthogonal code in the lookup table which is associated with the minimum count is the closest match for the corrupted received code. The matched orthogonal code in the lookup table is the corrected code, which is then decoded to a-bit data. The receiver is able to correct up to (b/4)-1 bits in the received impaired code. However, if the minimum count is associated with more than one combination of orthogonal code then a signal, REQ, goes high.

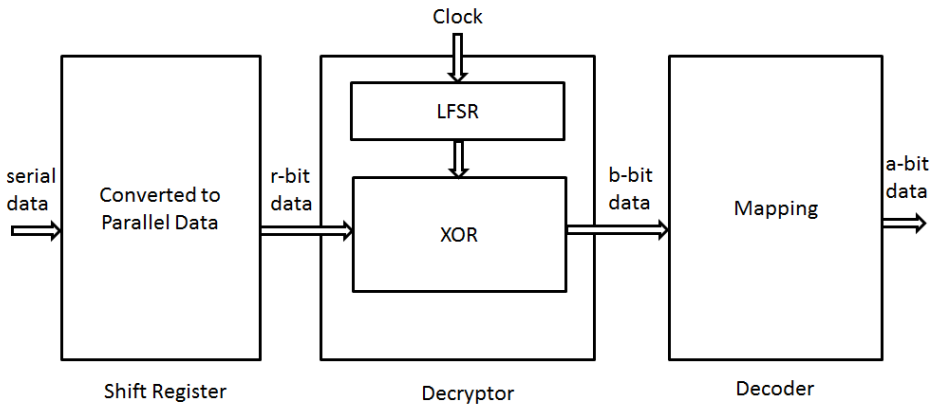

Figure 2. Block diagram of receiver

4. RESULTS AND ANALYSIS

In order to test the code ISE Xilinx software and a hardware board of Spartan-3 were used. Modelsim XE software is used to perform simulation. The output of the simulation or the final results is checked for most of the 5-bit combinations of input and 16-bit orthogonal code. The process of software simulation along with the working of clock cycles is briefed further for both transmitter and receiver.

Transmitter:

The internal process of transmitter and the simulation results can be observed in figure 3. The 5-bit t_data input signal for the encoder is 01001 where the orthogonal code works on it and a 16-bit output i.e. b=5 orthogonal code output $2^{b-1} = 2^4 = 16$ is obtained. The t_ortho signal as shown in the figure 3 represents the output of the encoder "AAAA H" (1010101010101010) which is fed as input to the encryptor. The output of the encryptor t_out "AAAB H" is obtained for the respective input. To enable the transmission of serial bit data reset signal is used for every rising edge of the clock.

Receiver:

Once the data is obtained at the receiver the serial data is transformed into parallel data. The r_data signal represents the input signal of receiver. This data is decrypted and an orthogonal code is obtained which is represented by r_ortho signal. This data acts as an input to the decoder unit. A counter variable is used which counts the number of 1's in the result obtained, when the received code is XOR'd with all the possible combinations of orthogonal code. The original data is obtained by checking the minimum count of the received data. There would be 4 cases for all the simulation results available. In the first case the received data r_data=1010101010101011. The r_data is given as the input to the decryptor, the output of decryptor is r_ortho=1010101010101000. It is given as input to the decoder, the decoder checks for the closest match to the r_ortho by performing XOR operation between r_ortho and each and every value in the lookup table and gives the minimum count=00000. The value at the signal count represents number of errors presented in the received data. In this case number of errors is zero. The value associated with the minimum count is the original data is represented by a signal r_out=01001 as shown in the figure 4.

In the second case the received data r_data=1000101010101011. The r_data is given as the input to the decryptor, the output of decryptor is r_ortho=8aacH. It is given as input to the decoder, the decoder checks for the closest match to the r_ortho by performing XOR operation between r_ortho and each and every value in the lookup table and gives the minimum count=00010. In this case number of errors is two. The value associated with the minimum count is the original data is represented by a signal r_out=01001 as shown in the figure 5.

In the third case the received data r_data=1001101010101011. The r_data is given as the input to the decryptor, the output of decryptor is r_ortho=1001101010101100. It is given as input to the decoder, the decoder checks for the closest match to the r_ortho by performing XOR operation between r_ortho and each and every value in the lookup table and gives the minimum count=00011. In this case number of errors is three. The value associated with the minimum count is the original data is represented by a signal r_out=01001 as shown in the figure 6.

In the fourth case the r_data= 1001110010101011 is obtained as input. But there is no closest match obtained for the respective orthogonal code and the count value is more than the number of errors produced. In this case the output is not obtained and the r_req goes high which requests the sender to resend the message.

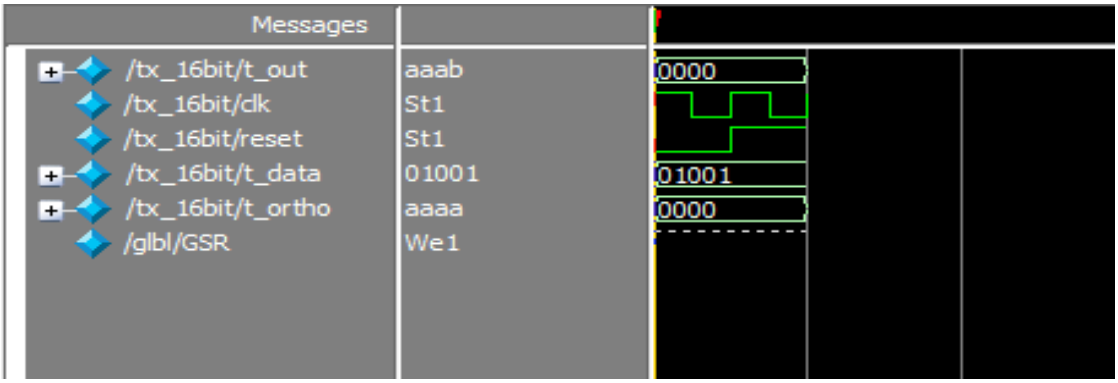

Figure 3. Simulation result of transmitter with encryption

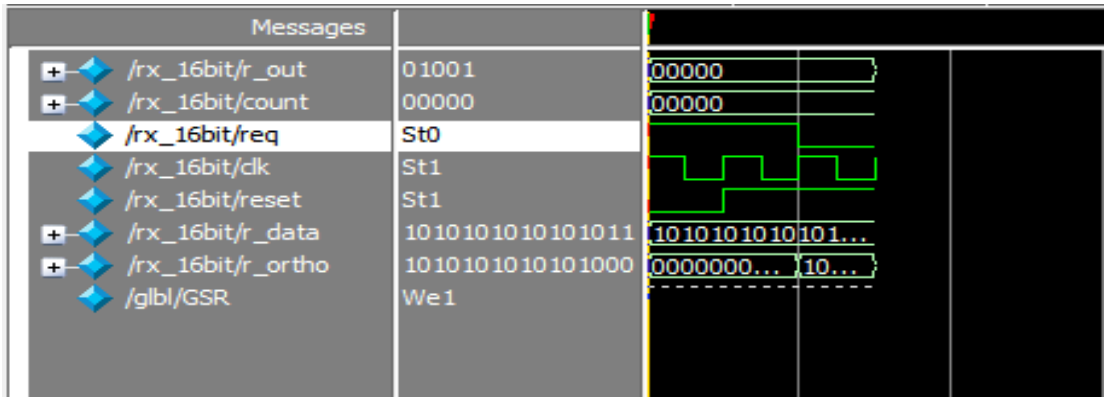

Figure 4. Simulation result of receiver with zero errors

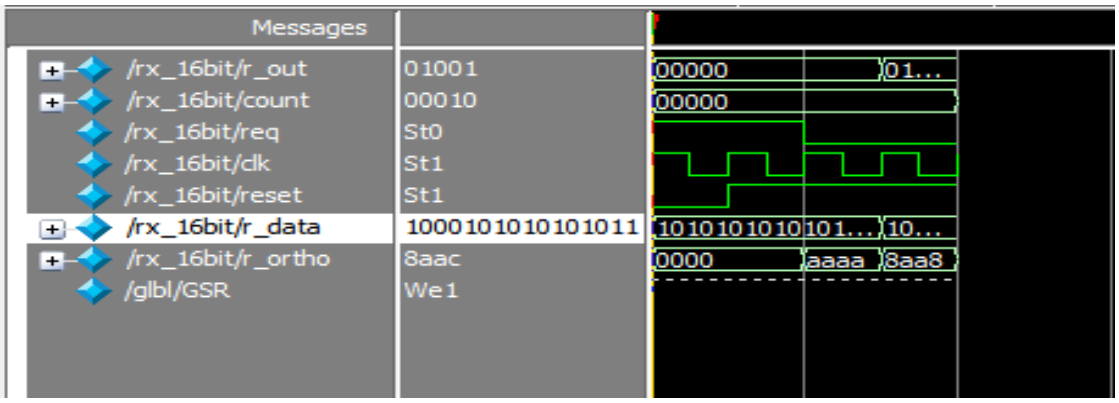

Figure 5. Simulation result of receiver with 2-bit error

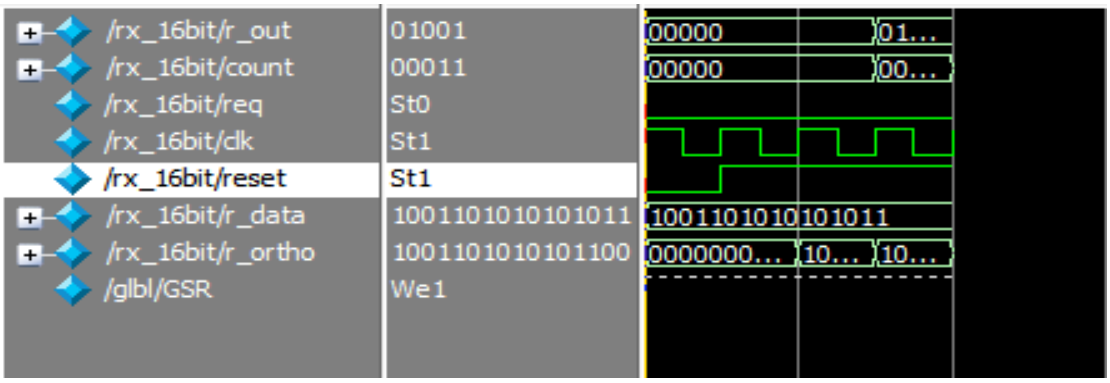

Figure 6. Simulation result of receiver with 3-bit error

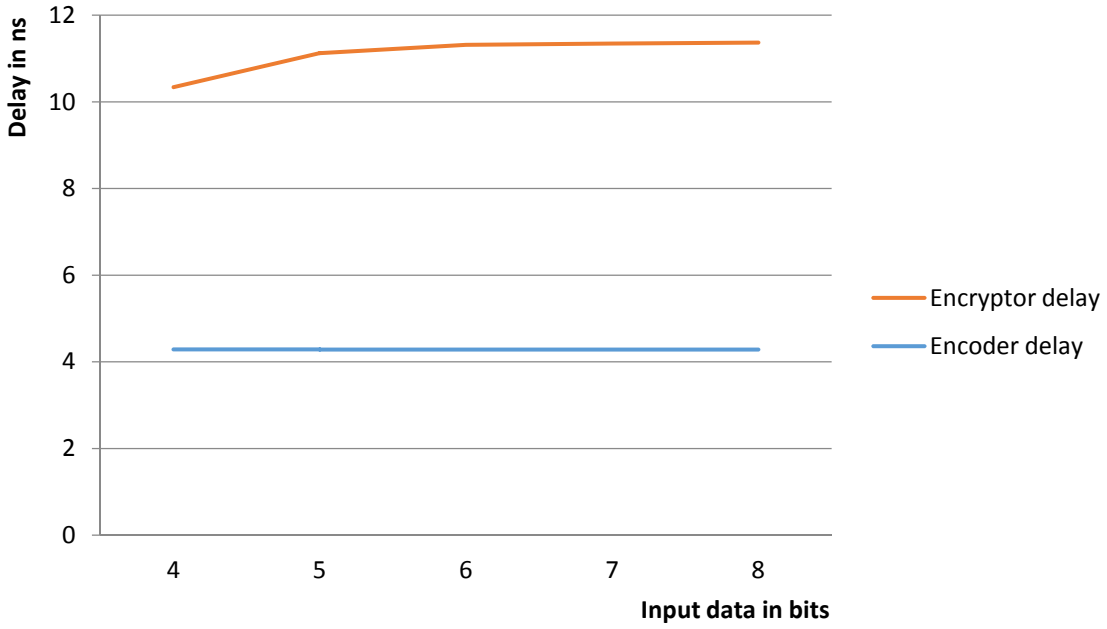

Figure 7. Input data in bits Vs encoder & encryptor delay

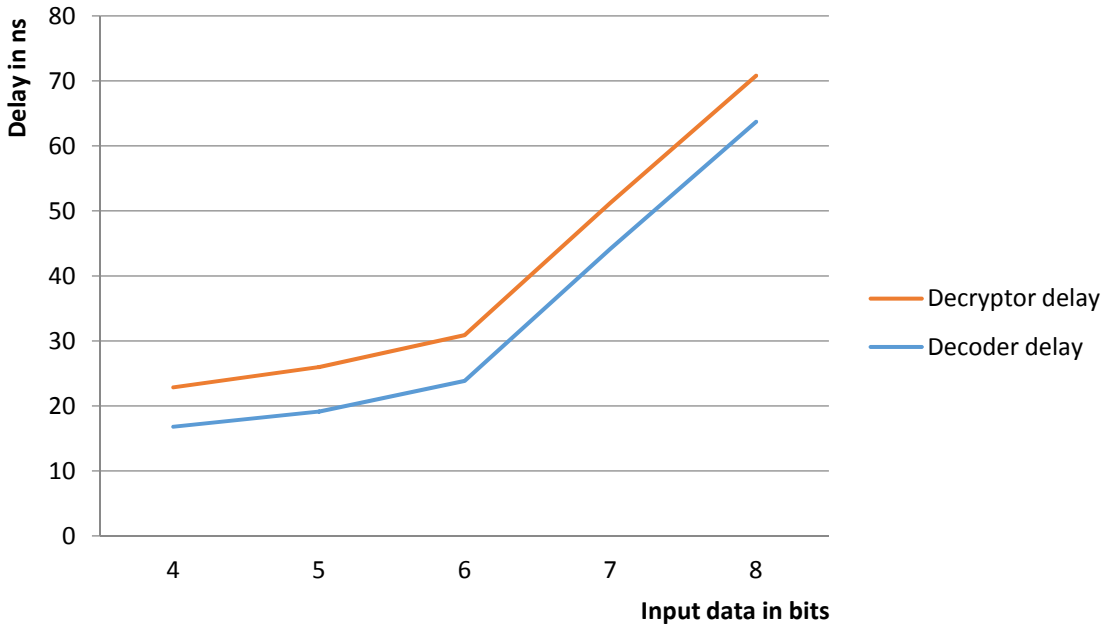

Figure 8. Input data in bits Vs decoder & decryptor delay

The simulation results show that for an a-bit data encoded into b-bit orthogonal data and is encrypted into a cipher text. A varying number of orthogonal code combinations are obtained which is able to detect any faulty combination. 2^a is the total number of orthogonal code combinations obtained. The error detection and correction percentage = $(2^b\text{-}2b)/2^b$. The total system can detect and correct till (b/4)-1 bit error and similarly the number of clock cycles necessary for the data received to get processed is (2b+2). Consider an example like 5-bit data therefore the total number orthogonal code bit length is $2^b=2^{5-1}=16$, and the total number of orthogonal code combinations are 2^{16}. In order to encrypt the data, the key length should be of the same size as that of the data obtained from the encoder i.e. 16-bit key is required to encrypt the data. Hence

each 5-bit data has a unique 16-bit orthogonal code so a 16-bit key required encrypting that data. The 16-bit key is obtained using a pseudo random generator LFSR. The delay time for encryption and decryption is 6.840ns when the input bit length is 5, similarly when the input is a 6-bit data the delay times are 7.042ns which shows that there is an increase in the delay times as the input data increases. So, as the input data increases the orthogonal bit length also increases which results in more security as there is an increase of key length to encrypt the data. There by as input increases the encryption time also increases for individual systems and the synthesis reports for the delay time are shown in Table II. The percentage of error correction and detection obtained is $(2^{16}-2*16)/2^{16}$ =99.95% with error correcting ability and the number errors it can detect is 3. Similarly the error correction and detection percentage for 32 bit orthogonal code is 99.99% and the number of errors that can be detected are 5. Hence the possible number of combinations received at the receiver will be able to detect the correct code with the orthogonal code available.

Table 2 shows the error correction and detection rates from the simulation results obtained for 4-bit, 5-bit, 6-bit, 7-bit, 8-bit data as input. Table 3 shows the delay timings for both transmitter and receiver along with the internal block delays obtained during synthesis. We can observe the as the input bit length and the key length to encrypt the data increases the delay times for the individual systems also increases.

Table 2. Summary of results and their error correction and detection results

Input Bits (a)	Ortho code output bit length (b)	No of errors(t)	% of errors detected and corrected
4	8	1	93.75
5	16	3	99.95
6	32	7	99.99
7	64	15	99.99
8	128	31	100
A	b=2^(a-1)	t=(b/4)-1	%=(2^b-2^b)/2^b

Table 3. Summary of transmitter and receiver delay times obtained from the synthesis reports for 4-bit, 5-bit, 6-bit, 7-bit and 8-bit

Input data in bits	Encode delay(ns)	Encryptor delay(ns)	Transmitter delay(ns)	Decryptor delay(ns)	Decoder delay(ns)	Receiver delay(ns)
4	4.283	6.054	6.054	6.054	16.786	14.898
5	4.283	6.840	6.840	6.840	19.133	18.766
6	4.283	7.032	7.032	7.032	23.853	24.089
7	4.283	7.062	7.062	7.062	44.163	45.478
8	4.283	7.082	7.082	7.082	63.718	64.238

5. CONCLUSION

The results of the present work show that the error detection and correction rate has been increased to 100% when the input length is an 8-bit data. The encryption and decryption time delays are also increased when the input data is increased as observed from the results of 4-bit, 5-bit, 6-bit, 7-bit and 8-bit data as input. Future work includes maintaining constant time delays for varying input length by using crypt analysis techniques and band width limitation.

REFERENCES

[1] Reviriego P, Pontarelli S, “Reducing the Cost of Single Error Correction with Parity Sharing”, Vol: 13, Issue: 3, pp: 420-422, IEEE Transactions-2013.
[2] Rukmani R, “Error Detection and Correction Architecture for Motion Estimation in video coding systems”, pp:1-5, IEEE Conference-2013
[3] Jayarani M.A, Jagadeeswari M, “A novel fault detection and correction technique for memory applications”, pp: 1-6, IEEE Conference ICCCI-2013.
[4] Anton C, Ionescu L, “Error detection and correction using LDPC in parallel Hopfield networks”, pp: 1-4, IEEE Conference ISEEE-2013.
[5] Saiz-Aadalid, Gil p, “Modified hamming codes to enhance short burst error detection in semiconductor memories”, pp: 62-65, IEEE Conference EDCC-2014.
[6] Baskar S, “Error recognition and correction enhanced decoding of hybrid codes for memory application”, pp: 1-6, IEEE Conference ICDCS-2014.
[7] Bo Dai, Zhensen Gao, “Orthogonal DPSK/CSK Modulation and Public Key Cryptograpgy Based Secure Optical Communications”, vol: 25, Issue: 19, pp: 1897-1900, IEEE-2013.

[8] Lamonica M, "Long-Distance quantum cryptography", vol: 50, Issue: 8, pp-12-13, IEEE Journals and magazines-2013.
[9] Sukalyan Son, Sayani Sen, "A Non-adaptive Partial Encryption of Grayscale Images Based on Chaos", Science Direct, Procedia Technology, 10, pp 663-671, CIMTA-2013.
[10] Hossein Rahmani, Elankovan Sundararajan, "Encryption as a Service (Eaas) as a Solution for Cryptography in Cloud", Science Direct, Procedia Technology, 11, pp 1202-1210, ICEEI-2013.
[11] Marius Iulian, "New Enrollment Scheme for Biometric Template using Hash Chaos-Based Cryptography", Science Direct, Procedia Engineering 64, pp: 1459-1468, 2013.
[12] Xiaotian Wu, Wei Sun, "Entended Capabilities for XOR-based visual cryptography", vol: 0 Issue: 10, pp: 1592-1605, IEEE journals-2014.
[13] Syed Rizvi, Katie Cover, "A Trusted third party based Encryption Scheme for Ensuring data confidentiality in cloud environment", Science direct, pp: 381-386, 2014.
[14] Xuanxia Yao, Zhi Chen, "A Light weight attribute-based encryption scheme for the internet of things", Elsevier-2014.
[15] Charles H, Gills Brassard, "Quantum cryptography: public key distribution and coin tossing", 2014.
[16] Zhang Qiming, "Secure Digital Certificate Design based on the Public Key Cryptography Algorithm", vol: 11, No. 12, pp 7366-7372, 2013.
[17] P.K. Das, "Bounds on Codes Correcting Periodic Errors Blockwise", vol: 2, No.1, pp 51-56, International Journal of Informatics and Communication Technology-2013.