Implementing a Hybrid Crypto-coding Algorithm for an Image on FPGA

B.V. Srividya[1(✉)] and S. Akhila[2]

[1] Department of Telecommunication Engineering,
DayanandaSagar College of Engineering, Bangalore, Karnataka, India
srividyabv@gmail.com
[2] Department of Electronics and Communication Engineering,
BMS College of Engineering, Bangalore, Karnataka, India
akhila.ece@bmsce.ac.in

Abstract. This paper proposes a hardware design, implemented on an FPGA, for a hybrid selective encryption and selective error correction coding scheme. FPGA's are used as implementation platforms in image processing, as its structure exploits the temporal and spatial parallelism. The algorithm aims at implementing security and reliability in which encryption and encoding are performed in a single step using Bezier curve and Galois field GF (2^m). The system aims at speeding up the encryption and encoding operations without compromising either on security or on error correcting capability by using selective encryption and selective encoding. The coding for hybrid crypto-coding algorithm is carried out using VHDL. The algorithm is simulated and synthesized using Xilinx ISE 10.1 software. The algorithm is implemented on Spartan 3 FPGA device 3s1000fg676-5. The proposed scheme reduces the hardware as modular arithmetic operations are involved.

Keywords: Bezier curve · Galois field · Image · Encryption · Error correction · FPGA

1 Introduction

In order to obtain a high throughput rate in Image processing, the algorithms are implemented in Field Programmable Gate Array (FPGA), which is a reconfigurable hardware. Implementing on FPGA provides low power cost effective solution and a high data throughput.

Traditionally HDL languages such as VHDL and Verilog are used for implementing on FPGA. In this paper, using VHDL an image is encrypted using the concept of selective encryption that is based on Quartic Bezier Curve over Galois Field GF (2^m). Further the encrypted image is recovered from transmission errors using Low Density Parity Check Codes (LDPC). The hybrid crypto-coding algorithm is implemented on FPGA.

The following sections give a brief introduction to Bezier curves and Galois Field based on which the encryption and the error recovery algorithm are constructed and

S.C. Satapathy and A. Joshi (eds.), *Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 2*, Smart Innovation, Systems and Technologies 84, DOI 10.1007/978-3-319-63645-0_8

implemented on FPGA. The Low Density Parity Check codes is also been discussed in the following section.

A. FPGA Overview

An FPGA is made up of an array of programmable logic cells that are interconnected using a network of interconnecting lines with switches amidst them. The reconfigurable interconnects allows the logic cells to be interconnected, thereby configuring the logic cells to perform the desired logical operations. Around the boundary of the chip, Input Output Cells exist. These I/O cells provide an interface between the external pins of the chip and the interconnecting lines. Indicating the logic function for each cell and for the switches is termed as programming an FPGA.

B. Introduction to Bezier Curves

Bezier curves are a method of designing polynomial curve segments [1, 2], where the shape of curves can be controlled using the control points. The control points (from P0 to Pn) of the Bezier curve determine the order ‘n’ of the curve. Bezier Curves can be classified as linear Bezier curve, Quadratic Bezier curve, Cubic and Quartic Bezier curves on the basis of the order ‘n’ [3].

- A Linear Bezier curve has n = 1 and its curve equation is given by Eq. (1)

$$B(t) = (1-t)P_0 + tP_1, t \in [0, 1] \tag{1}$$

  where there are two control points P0 and P1. Linear Bezier curve represents an interpolation between two points.
- A Quadratic Bezier curve has n = 2 and the curve Eq. (2) is given by

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2, t \in [0, 1] \tag{2}$$

  where there are three control points P0, P1, and P2. The Quadratic Bezier curve represents a linear interpolate of the control points from P0 to P1 and also P1 to P2.
- The cubic Bezier curve is given by Eq. (3)

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3(1-t)t^2 P_2 + t^3 P_3; t \in [0, 1] \tag{3}$$

  where P0 to P3 are its control points.
- The Quartic Bezier curve B(t) having 5 control points from P0 to P4 is given by Eq. (4)

$$B(t) = (1-t)^4 P_0 + 4t(1-t)^3 P_1 + 6t^2(1-t)^2 P_2 + 4t^3(1-t)P_3 + t^4 P_4; t \in (0, 1) \tag{4}$$

C. Introduction to Galois Field

Evariste Galois is the inventor of Galois field. The number of elements is finite in GF (p^m). Some of the popular Forward Error Correcting codes like BCH Codes and Reed Solomon codes use finite fields for the purpose of encoding and decoding [4]. In cryptographic algorithms, the value of p is taken to be 2, and is represented as GF (2^m).

Every GF (2^m) has a primitive polynomial of degree m, which α and its conjugates to be its roots. From the primitive polynomial the elements of GF (2^m) can be constructed. The elements are {0, 1, α α^2, α^3… α^{m-2}}. Each element in GF (2^m) can be represented using m-bits. In coding theory and cryptographic algorithms, certain modular arithmetic operations are performed on the elements of the Galois Field. The following section shows the construction of the elements of the field and the arithmetic operations performed on the field elements.

Table 1. Elements of GF (2^4)

Element	Polynomial representation	Binary representation
0	0	(0000)
α^0	1	(1000)
α^1	X	(0100)
α^2	X^2	(0010)
α^3	X^3	(0001)
α^4	X + 1	(1100)
α^5	$X^2 + X$	(0110)
α^6	$X^2 + X^3$	(0011)
α^7	$1 + X + X^3$	(1101)
α^8	$1 + X^2$	(1010)
α^9	$X + X^3$	(0101)
α^{10}	$1 + X + X^2$	(1110)
α^{11}	$X + X^2 + X^3$	(0111)
α^{12}	$1 + X + X^2 + X^3$	(1111)
α^{13}	$1 + X^2 + X^3$	(1011)
α^{14}	$1 + X^3$	(1001)

Table 2. Addition in GF (2^4)

+	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14
2	2	3	0	1	6	7	4	5	10	11	8	9	14	15	12	13
3	3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12
4	4	5	6	7	0	1	2	3	12	13	14	15	8	9	10	11
5	5	4	7	6	1	0	3	2	13	12	15	14	9	8	11	10
6	6	7	4	5	2	3	0	1	14	15	12	13	10	11	8	9
7	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8
8	8	9	10	11	12	13	14	15	0	1	2	3	4	5	6	7
9	9	8	11	10	13	12	15	14	1	0	3	2	5	4	7	6
10	10	11	8	9	14	15	12	13	2	3	0	1	6	7	4	5
11	11	10	9	8	15	14	13	12	3	2	1	0	7	6	5	4
12	12	13	14	15	8	9	10	11	4	5	6	7	0	1	2	3
13	13	12	15	14	9	8	11	10	5	4	7	6	1	0	3	2
14	14	15	12	13	10	11	8	9	6	7	4	5	2	3	0	1
15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

(i) ***Elements in GF (2^m)***

The elements of Galois field GF (2^4) is constructed using the primitive polynomial P (x) = $x^4 + x + 1$ and is shown in Table 1

(ii) ***Addition in GF (2^m)***

Galois field addition is explained with an example: The primitive polynomial of Galois Field GF (2^4) is P(x) = $x^4 + x + 1$. This primitive polynomial has α and its conjugates as the roots.

According to Table 1, each element of GF (2^4) is represented using 4-binary bits. Addition is performed using Bitwise XORing operation. For example:

$$\alpha^5 + \alpha^5 = (0110) + (0110) = (0000) = 0 = \alpha^0$$
$$\alpha^2 + \alpha^5 = (0010) + (0110) = (0100) = 1 = \alpha^1$$

There is a significant increase in the speed of addition, as there is no carry generation and carry propagation delay.

The addition table for the same is as shown in Table 2.

(iii) ***Multiplication in GF (2^m)***

Modular multiplication is performed, by multiplying the polynomials and then performing modular reduction on the product. Let a(x), b(x) be the polynomial representation of two elements in GF (2^m), whose product needs to be computed and g(x) be the irreducible field generator polynomial, then modular multiplication is as illustrated in the following example.

Example: If g(x) = 1 + X + X^4, a(x) = 1 + X^3, b(x) = 1 + X^2

Then a(x) * b(x) = (1 + X^3) * (1 + X) = (1 + X^2 + X^3 + X^5)

Modular reduction of the above result is (1 + X^2 + X^3 + X^5) mod (1 + X + X^4) = X^3 + X + 1.

Table 3 is the modular multiplication performed on the elements of GF (2^4) using the polynomial g(x) = 1 + X + X^4

Table 3. Multiplication in GF (2^4)

X	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	2	4	6	8	10	12	14	3	1	7	5	11	9	15	13
3	3	6	5	12	15	10	9	11	8	13	14	7	4	1	2
4	4	8	12	3	7	11	15	6	2	14	10	5	1	13	9
5	5	10	15	7	2	13	8	14	11	4	1	9	12	3	6
6	6	12	10	11	13	7	1	5	3	9	15	14	8	2	4
7	7	14	9	15	8	1	6	13	10	3	4	2	5	12	11
8	8	3	11	6	14	5	13	12	4	15	7	10	2	9	1
9	9	1	8	2	11	3	10	4	13	5	12	6	15	7	14
10	10	7	13	14	4	9	3	15	5	8	2	1	11	6	12
11	11	5	14	10	1	15	4	7	12	2	9	13	6	8	3
12	12	11	7	5	9	14	2	10	6	1	13	15	3	4	8
13	13	9	4	1	12	8	5	2	15	11	6	3	14	10	7
14	14	15	1	13	3	2	12	9	7	6	8	4	10	11	5
15	15	13	2	9	6	4	11	1	14	12	3	8	7	5	10

Another approach for performing modular multiplication, when the elements of the field are represented in binary values are as explained below.

The Binary representation of g(x) = (X^4 + X + 1) = (1101).

The modular multiplication in binary can be performed as illustrated in Table 3.

For example: If A = 9 and B = 9, then

$$\mathrm{AXB} = 9 \times 9 = (1001) \times (1001) = (1010001)$$

$$(1010001) \bmod (1101) = (1011) = 13$$

Further, exponential operation can be performed using GF (2^m) as shown below.

$$
\begin{aligned}
5^7 &= (5 \times 5 \times 5 \times 5 \times 5 \times 5 \times 5)\mathrm{GF}(2^4) \\
&= (2 \times 2 \times 2 \times 5)\mathrm{GF}(2^4) \\
&= (4 \times 10)\mathrm{GF}(2^4) \\
&= 14
\end{aligned}
$$

D. Low Density Parity Check Codes (LDPC)
Low density parity check codes fall into the category of linear block codes and is one of the popular error correcting codes, when data is transmitted over a noisy channel. The density of one's is smaller compared to that of zeros in LDPC [14]. There can be a regular or an irregular Parity matrix defined for an LDPC code. If the Parity matrix has a uniform row and column weight, then it is a Regular parity matrix P [16]. Every row and every column of the Regular Parity matrix has exactly the same number of elements. These conditions ensure that the parity matrix P has uniform row and column weights forming a Regular LDPC code. The Parity matrix P that does not adhere to the property of having uniform row and column weight forms an Irregular Parity matrix [16].

2 Related Work

The work on "Joint AES algorithm and LDPC codes" [5] by CP Gupta et al. discusses on achieving Security and error correction in a single step as, AES is secured and also LDPC codes retains full error correction capability. But, in symmetric key cryptosystems the two parties who are communicating need to share the secret key prior to the start of the transaction.

The authors of "Joint Encryption and Error Correction Technical Research Applied an Efficient Turbo Code" [6] Jianbin Yao et al., is effective in terms of security and reliability. But the system has not been verified for many attacks. The image recovery is achieved after several iterations.

The authors of "Implementation of High Security Cryptographic System with Improved Error Correction and Detection Rate using FPGA" [7] have discussed on achieving 100% Error detection, encryption scheme is effective and bandwidth is improved. But, the encryption and decryption delays increase as the input data is increased from 4-bits to 8-bits.

The proposed algorithm is on combining selective encryption and selective encoding to obtain an secured error free data. The encryption algorithm is a public key cryptosystem, where in encryption [8] and decryption [8] operations are performed with a pair of mathematically related keys [9, 10] based on the Galois Field GF (p^m). In performing, Selective Encoding the complexity of the hardware is simplified, a better performance of the decoder is achieved even when output is zero. Further there is a reduction in the area as the Hardware used is XOR gates. The algorithm uses, K-P modulo-additions for decoding, where K being the length of the information digits and P is the number of non zero digits of the Parity matrix.

The following Sect. 3 discusses on the proposed system, Sect. 4 is on results and discussion. Section 5 discusses the conclusion arrived for the proposed system.

3 Proposed Work

The structure of the proposed encryption and encoding scheme is as shown in Fig. 1.

Security is achieved using the encryption algorithm as discussed in [20].

For correcting the errors, the modified Low density parity check codes have been explained in [21].

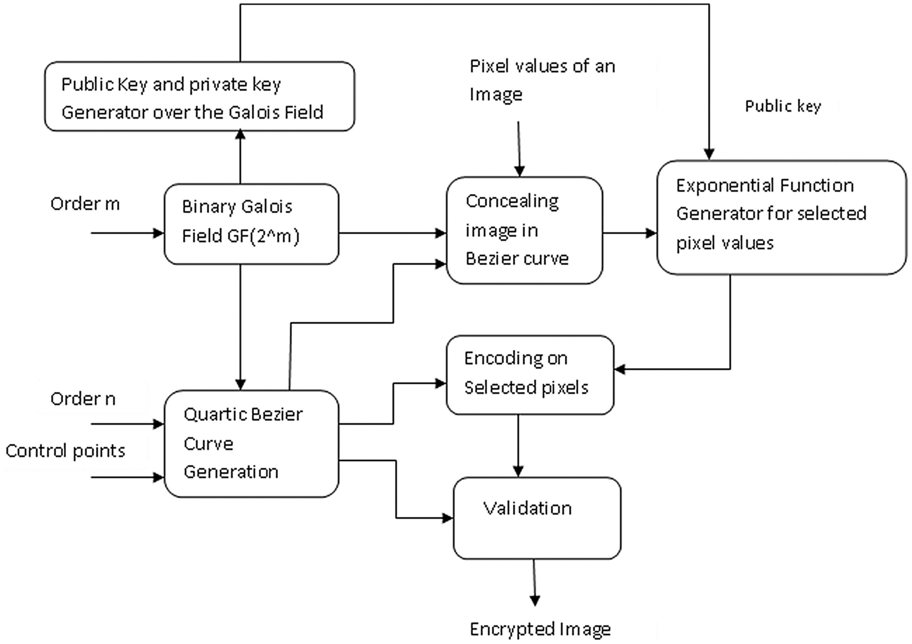

Public Key and private key Generator over the Galois Field
Pixel values of an Image
Public key
Order m
Binary Galois Field GF(2^m)
Concealing image in Bezier curve
Exponential Function Generator for selected pixel values
Encoding on Selected pixels
Order n
Control points
Quartic Bezier Curve Generation
Validation
Encrypted Image

Fig. 1. Encryption and encoding scheme

A secured error free image can be obtained as explained according the following algorithm:

- The control points, of the n-order Bezier curve and the m-order finite field are shared between the sender and the receiver before the start of the transaction. Generate the Bezier image from the n-order Bezier curve. This image has the same size as that of the data image. The data image is concealed in the Bezier image as shown in the Fig. 2, and is denoted as I. This image I is further selectively encrypted as explained in the following section.

- **Selective Encryption:**
  The Selected Pixel values of the aggregated image I is exponentially raised to the power of m^e, over GF (2^m) as given by the relation (5)

$$C1 = (I)^{m^e} \bmod GF(2^m). \tag{5}$$

  I is the selected pixel values of the concealed image. The order of the finite field ‘m’ is the secret key and the public key is ‘e’. Only pixel values greater than the threshold value are encrypted resulting in selective encryption. The 2^m digits which are selectively encrypted are selectively encoded as explained in the following section.
- **Data Reliability using Selective Encoding:**
  The Data Reliability is achieved by the construction of LDPC codes based on n-order Bezier Curve over Galois Field GF (2^m) [13, 15] to obtain full error correcting capability. LDPC codes have a better performance when combined with Galois Field [17, 18]. Non-Binary LDPC codes is a better choice when more number of errors needs to be corrected [13, 19]. The data that needs to be selectively encoded is the selectively encrypted 2^m digits. If this 2^m digits of selectively encrypted data has two consecutive digits data (i) and data (i + 1) to be the same, then the first digit data (i) is replaced with a zero. This process continues till all the 2^m digits of data have been checked for repetition with its adjacent value. This selected data denoted as Cs has zeros when adjacent values are the same. This selected data Cs is encoded, by performing modular multiplication of Cs and the Generator matrix G.

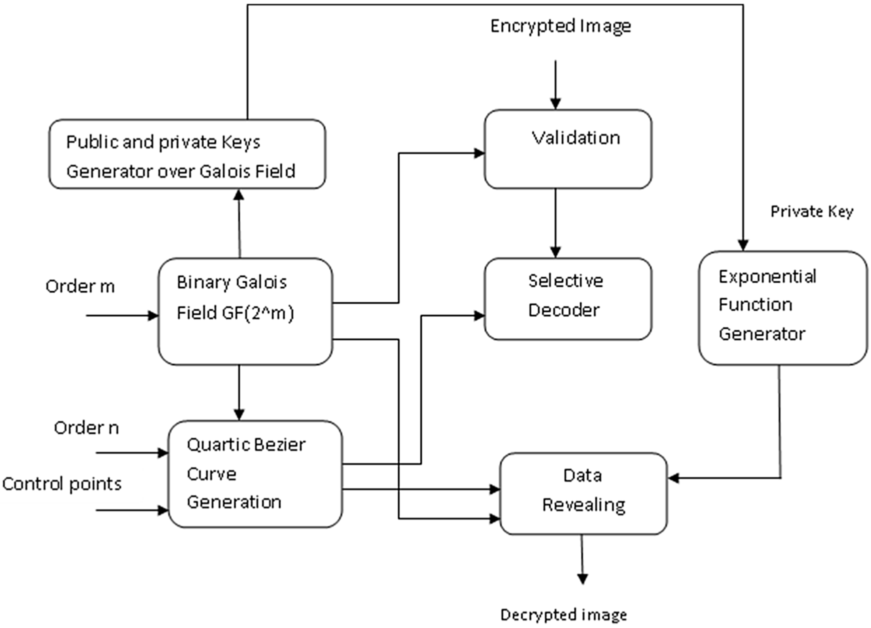

Encrypted Image
Public and private Keys Generator over Galois Field
Validation
Private Key
Order m
Binary Galois Field GF(2^m)
Selective Decoder
Exponential Function Generator
Order n
Quartic Bezier Curve Generation
Control points
Data Revealing
Decrypted image

Fig. 2. Decryption and decoding scheme

- **Validation at transmitter:**
  The selectively encoded data CE is converted to CE1 after validation.
  Mean of P1, P2, P_{n-1} is computed, where P1, P2,
  P_{n-1} are the control points on the curve. P_0 and P_n are the starting point and the ending point of the n-order Bezier curve respectively.

$$\Delta = (xt, yt) = \frac{(P_n - mean)}{P_n - P0} \tag{6}$$

  Defined over the order 'm' of the Galois field. This value of Δ, as shown in Eq. (6) is used for the purpose of validation.

$$CE1 = \frac{\Delta}{CE} \bmod \; GF(2^m) \tag{7}$$

  Where CE1 as shown in Eq. (7), is the 2^{m+1} digits of cipher text and CE is the selectively encoded value.

Decryption with Decoding embedded:
Figure 2 shows the Decryption of the cipher text with a decoder embedded to retrieve the plain text.

The receiver upon receiving every 2^{m+1} digits of cipher text of an image, checks for the authentication of the data using the value of Δ, calculated using the control points. After checking for validation, the modified LDPC decoder makes the received data to be error free. The error free data is exponentially raised to $(m^{-1})^d$ over GF (2^m), with 'd' being the private key. These exponentially raised pixel values are algebraically combined with the Bezier image by the receiver to obtain the data image.

- **Validation at receiver:** The receiver calculates the selectively encoded value using Eq. (8).

$$CE = \frac{\Delta}{CE1} \bmod \; GF(2^m) \tag{8}$$

  where CE1 is the 2^{m+1} digits of cipher text and CE is the selectively encoded value. The value of Δ is used for the purpose of validation. The value of Δ is derived from the control points of the chosen n-order Bezier curve. If the value of Δ is not known to the receiver, then the erroneous information cannot be corrected, thereby preserving the security of the information.
- **Selective decoding:**
  After validation is performed, the received vector R contains checksum which is 2^m digits and the erroneous pixel values which is 2^m digits. The decoder corrects the image from the received erroneous image and makes it error free. To achieve this, the FEC codes are applied. In LDPC, each row of the encoded image has 2^{m+1} digits of data that is given as input to the decoder. Syndrome is calculated by the decoder to determine the position of errors in the received information. The syndrome S is a modular multiplication of the Received data R and the parity check matrix H. If the

Syndrome S is Zero, then the received vector is error free else, the decoder determines the location of the error. The error location is determined by referring to the Parity Check matrix H. The erroneous data is corrected using the appropriate checksum. After correcting the errors, the consecutive zeros will be replaced by the right most non zero pixel value.

- **Selective Decryption:**

  The Error free image V is selectively decrypted using the following logarithmic equation, to obtain the concealed image I, as shown in Eq. (9).

$$I = (V)^{(m^{-1})^{d}} \bmod GF(2^{m}) \tag{9}$$

where ‘m^{-1}’ is the inverse of the secret key and the private key for decryption is ‘d’. In public key cryptosystem, the pairs of keys used for encryption and decryption are related mathematically. The relation between the public key e, that is used for encryption and the private key d that is used for decryption is given by Eq. (10)

$$m^{e}(m^{-1})^{d} = 1 \bmod GF(2^{m} - 1) \tag{10}$$

- **Data Revealing:**

  The original image is embedded in the Bezier image. This concealed image is selectively encrypted, and after error recovery, the same is selectively decrypted. After selective decryption, the original image is retrieved from the concealed image.

4 Results and Discussion

The proposed hybrid crypto-coding algorithm is coded in MATLAB and also in VHDL.

The image considered for experimental purpose is lena.jpg which is of size 256 × 256 pixels.

The results are obtained are using Quartic Bezier curve and Galois field GF (2^8).

The results are discussed for an encrypted image that is affected with White Gaussian Noise having SNR = 0.5 dB

The following Fig. 3 is a snapshot of the data image, the encrypted image, the received erroneous image and the decrypted image from MATLAB.

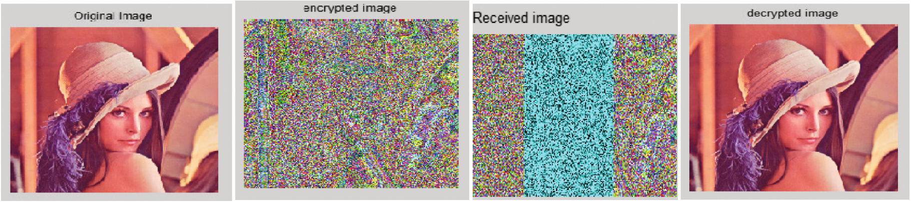
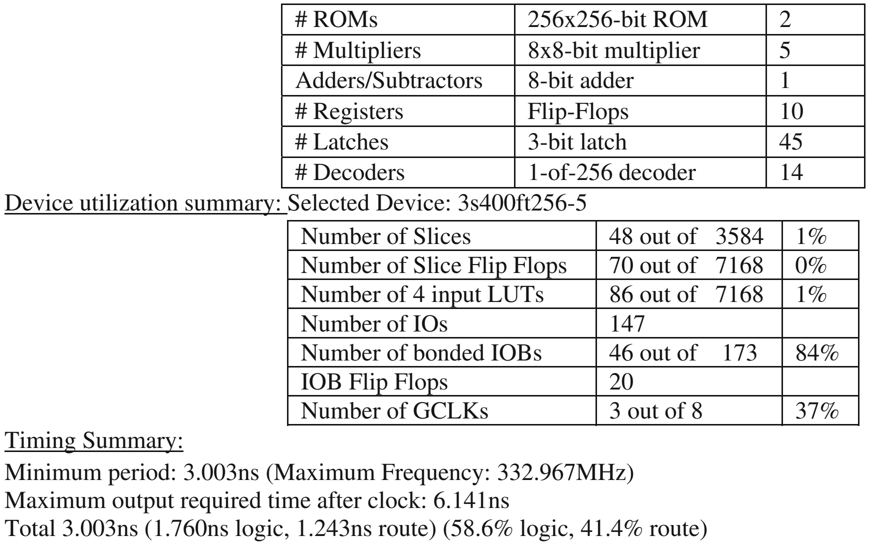


Fig. 3. (a) Original image (b) Encrypted image (c) Received image (d) Decrypted image

From Fig. 3, it can be observed that the received image which has been encrypted and transmitted has been modified due to the presence of Gaussian noise. This modified image has non-zero syndrome values from S1 to S256. Only the authenticated user can correct these modified pixel values. After the valid authentication check the errors have been eliminated using the decoding algorithm. It can be seen that the Decrypted image is same as the original image. 39,322 modified pixel values have been detected and corrected using the proposed algorithm.

The coding for hybrid crypto-coding algorithm is also carried out in VHDL, simulated and synthesized using the Xilinx ISE. The algorithm is implemented on FPGA Spartan 3 3s400ft256-5. The synthesis results obtained are shown in Fig. 4.



# ROMs	256x256-bit ROM	2
# Multipliers	8x8-bit multiplier	5
Adders/Subtractors	8-bit adder	1
# Registers	Flip-Flops	10
# Latches	3-bit latch	45
# Decoders	1-of-256 decoder	14

Device utilization summary: Selected Device: 3s400ft256-5

Number of Slices	48 out of 3584	1%
Number of Slice Flip Flops	70 out of 7168	0%
Number of 4 input LUTs	86 out of 7168	1%
Number of IOs	147	
Number of bonded IOBs	46 out of 173	84%
IOB Flip Flops	20	
Number of GCLKs	3 out of 8	37%

Timing Summary:

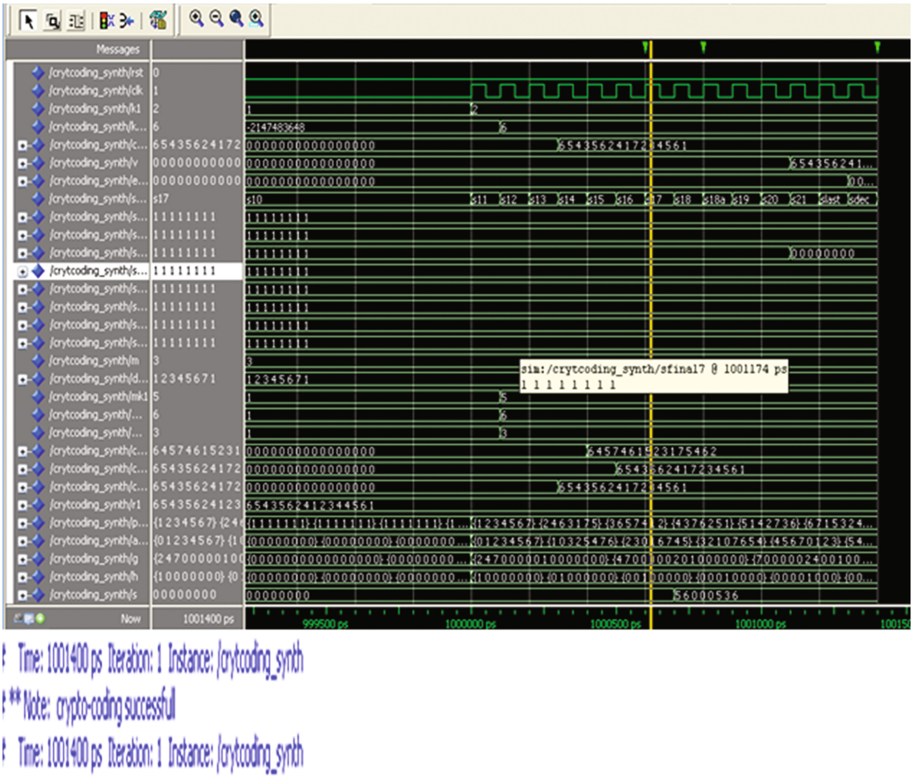
Minimum period: 3.003ns (Maximum Frequency: 332.967MHz)
Maximum output required time after clock: 6.141ns
Total 3.003ns (1.760ns logic, 1.243ns route) (58.6% logic, 41.4% route)

Fig. 4. Advanced HDL synthesis report

Figure 4 shows the hardware utilized when implemented on FPGA Spartan 3 3s400ft256-5. The timing summary indicates that 58% of the time is used for logic and 41% is used for routing.

The Fig. 5 shows the sample snapshot of the simulation results for combined selective encryption and selective encoding for an 8 × 8 pixel value of the original image using ModelSim.

Figure 5 shows that the crypto-coding algorithm is successful, as the encrypted data is made error free and then decrypted by the receiver.



```
# Time: 1001400 ps Iteration: 1 Instance: /crytcoding_synth
# ** Note: crypto-coding successful
# Time: 1001400 ps Iteration: 1 Instance: /crytcoding_synth
```

Fig. 5. Modelsim results of crypto-coding

5 Conclusion

This paper, establishes the working of combining Selective Encryption and selective Error Correction using modified LDPC with Bezier curve and Galois field GF (p^m). The Bezier curve points generated with the Galois field is the parity matrix P. This Parity matrix P is used for the construction of the generator matrix G and parity check matrix H. The original data image is concealed in the Bezier image by taking the aggregate of the pixel values. The pixel values of the concealed image above a certain threshold value are only encrypted. Thus Selective encryption is performed on the concealed image. Further, the proposed algorithm uses Selective Encoding, where in the repeating consecutive pixel values of the image are replaced by zeros. Using this approach, it is possible to encode only a few non-zero pixel values. The Encoding and Decoding involves modular arithmetic operations. The proposed decoder can handle BER = 1200/2048. The crypto-coding is done using one hot state encoding in VHDL and implemented on FPGA.

References

1. Caglar, H., Akansu, A.N.: A generalized parametric PR-QMF design technique based on Bernstein polynomial approximation. IEEE Trans. Sig. Process. **41**(7), 2314–2321 (1993)
2. [Online notes]: Bernstein; Visualization and Graphics Research Group. Department of Computer Science, University of California. www.idav.ucdavis.edu/education/CAGDNotes/Bernstein-Polynomials.pdf
3. Weisstein, E.W.: “Bézier Curve”, From MathWorld—A Wolfram Web Resource. http://mathworld.wolfram.com/BezierCurve.html
4. Cameron, P.J.: The Encyclopedia of Design Theory: Galois Fields 30 May 2003
5. Gupta, C.P., Gautam, S.: Joint AES algorithm and LDPC codes. Int. J. Sci. Eng. Res. **4**(7), 603–606 (2013)
6. Yao, J., Liu, J., Yang, Y.: Joint encryption and error correction technical research applied an efficient turbo code. Int. J. Secur. Appl. **9**(10), 31–46 (2015). doi:10.14257/ijsia.2015.9.10.03
7. Babu, N., Noorbasha, F., Gunnam, L.C.: Implementation of high security cryptographic systems with improved error correction and detection rate using FPGA. IAES Int. J. Electr. Comput. Eng. **6**(2), 602–610 (2016). ISSN 2088-8708
8. Stallings, W.: Cryptography and Network Security: Principles and Practice, 4th edn. Prentice-Hall Press, Upper Saddle River (2006)
9. Abdul Elminaam, D.S., Abdul Kader, H.M., Hadhoud, M.M.: Performance evaluation of symmetric encryption algorithms. Commun. IBIMA **8**, 58–63 (2009). ISSN 1943-7765
10. Jakhar, S.: Comparative analysis between DES and RSA algorithms. IJARCSSE **2**(7), 386–390 (2012)
11. Amounas, F., El Kinani, E.H., Hajar, M.: A novel approach for enciphering data based ECC using Catalan numbers. Inst. Adv. Eng. Sc. Int. J. Inf. Netw. Secur. (IJINS) **2**(4), 339–347 (2013). ISSN 2089-3299
12. Kute, V.B., Paradhi, P.R.: A software comparison of RSA and ECC. Int. J. Comput. Appl. **2** (1), 61–65 (2009)
13. Ganepola, V.S., et al.: Performance study of non-binary LDPC codes over Galois field. In: CSNDSP08. IEEE (2008)
14. Wasule, P.U., Ugale, S.: Review paper on decoding of LDPC codes using Advanced Gallagers algorithm. IJAICT **1**(7), 622–625 (2014)
15. AlinSindhu, A.: Galois field based very fast and compact error correcting technique. Int. J. Eng. Res. Appl. **4**(1), 94–97 (2014). www.ijera.com. ISSN 2248-9622 (Version 4)
16. Fossorier, M.P.C., Mihaljevic, M., Imai, H.: Reduced complexity iterative decoding of low density parity check nodes based on belief propagation. IEEE Trans. Commun. **47**(5), 673–680 (1999)
17. Chen, J.P., Fossorier, M.P.C.: Density evolution for two improved BP-based decoding algorithm for LDPC codes. IEEE Commun. Lett. **6**(5), 208–210 (2002)
18. Xu, M., Wu, J., Zhang, M.: A modified offset Min-sum decoding algorithm for LDPC codes. In: 3rd IEEE International Conference on Computer Science and Information Technology, (ICCSIT), vol. 3 (2010)
19. Kim, J., Ramamoorthy, A.: The design of efficiently encodable rate-compatible LDPC codes. IEEE Trans. Commun. **57**(2), 365–375 (2009)

20. Srividya, B.V., Akhila, S.: A heuristic approach for secured transfer of image based on Bezier curve over Galois field GF(p^m). In: IEEE International Conference on Circuits, Communication, Control and Computing (2014)
21. Srividya, B.V., Akhila, S.: Bezier curves with low density parity check codes over Galois field for error recovery in an image. In: IEEE International Conference on Communication and Signal Processing (2016)