Implementation of Data Security with Wallace Tree Approach Using Elliptical Curve Cryptography on *FPGA*

Thammaneni Snehitha Reddy, PG Student, Holy Mary institute of Technology And Science, Affiliated to JNTU, Hyderabad. Email: snehitha2206@gmail.com

Y. David Solomon Raju, Associate Professor, Holy Mary institute of Technology And Science, Affiliated to JNTU, Hyderabad. Email: davidsolomonraju.y@gmail.com

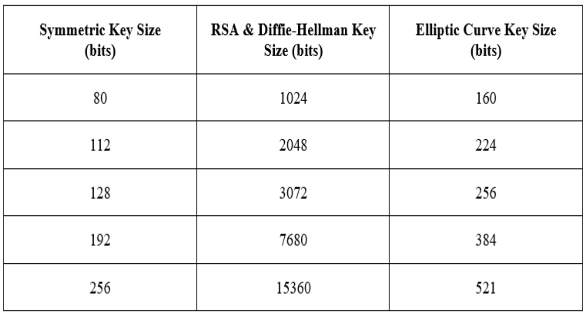
ABSTRACT

The growth of computing resources and parallel computing has led to significant needs for efficient cryptosystems over the last decade. Elliptic Curve Cryptography (ECC) provides faster computation over other asymmetric cryptosystems such as RSA and greater security. For many cryptography operations, ECC can be used: hidden key exchange, message encryption, and digital signature. There is thus a trade-off between safety and efficiency with regard to speed, area and power requirements. This paper provides a good ECC approach to encryption by replacing the Vedic multiplier (16 bit) with the Wallace tree multiplier with an improved output (128 bit). The proposed design processes data recurringly with less volume, less power consumption and greater velocity, in addition to improving efficiency. Using Xilinx 2015.2 software, the entire proposed design is synthesized and simulated and implemented on the ZYNQ FPGA Board. Compared with previous implementations, a significant improvement in field efficiency, time complexity and energy demand is demonstrated by the proposed design.

Keywords: Cryptography, Elliptic curve cryptography, Wallace tree multiplier, Area efficient, high speed

INTRODUCTION

Data protection refers to the defense against unauthorized data access and corruption over the life cycle of the data operation. The best data security cryptography provides two main areas, Cryptography symmetrical and asymmetric. ECC is the highest norm of asymmetric encryption (Elliptic Curve Cryptography). The Elliptic Curve Cryptography covers all relevant primitives of asymmetric cryptography such as digital signatures and key algorithms of agreement. The key ECC lengths linearly increase while the key lengths on existing RSA encryption methods exponentially increase with the increase in levels of security. For instance, 128-bit protection requires a 3,072-bit RSA key but only a 256-bit ECC key. The table 1 shows comparison between RSA and ECC for a given symmetric key size



Symmetric Key Size (bits)	RSA & Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table 1 : key size comparison

LITERATURE SURVEY:

This paper outlines the application of data security with cryptography, ECC.

Cryptography is a part of cryptology which is the art to make a cryptosystem capable of ensuring the security of facts. The true backup of digital documents is involved in cryptography. It refers to the design of mechanisms based entirely on mathematical algorithms which give crucial safety information. Cryptographic structures are typically divided into three different dimensions: Symmetric cryptography, Asymmetric cryptography, Hash functions.

Symmetric cryptography: A specific password/code or arbitrary number string or letter generated by a safe Random Number Generator may be the secret key for both sender and receiver to use

Asymmetric cryptography: The asymmetric variant is also referred to as public key cryptography. The use of private and public keys to encrypt and decrypt data is asymmetrical encryption. The keys are just large numbers paired but not identical (asymmetric).

Public key: A key that is shared with everyone is public key

Private Key: other key in the pair which is kept secret is called private key

Each key can be used for a message encryption; decryption uses the other key to the message encryption.

Hash functions: A hash function is a math feature that converts an input number into a numeric value that is compressed. The hash feature is randomly input, but the results are fixed.

Elliptic Curve Cryptography (ECC): Elliptical Curve Cryptography (ECC) is a form of public key encryption based on the theory of elliptical curves that can be used to construct faster, smaller and more precise cryptographic keys. ECC generates keys as a function of very large prime numbers rather than as the traditional generation mechanism by the characteristics of the elliptic curve equation. Combined with most public encryption strategies such as RSA and Diffie-Hellman, the technology can be used.

ENCRYPTION USING ECC ALGORITHM

Let' M' be the message we send. This message must be represented on the curve. Convert the original message to C1 and C2. The cipher text C1 is obtained by adding it to point P and by multiplying the added point with k.

Consider the' E' curve, randomly select' k' from [1–(n-1)].random number k (64 bit) Two cipher texts will be generated so that they are C1 and C2.

Text cipher c1= k*p ……. (2)

Figure 1 shows the Wallace tree 64 bit multiplier for c1

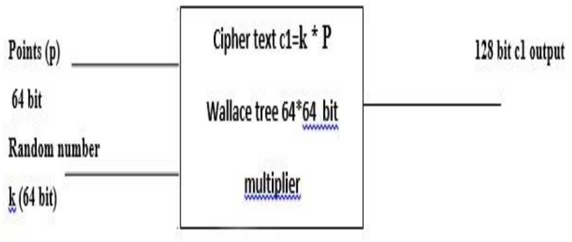


Figure 1: Wallace Tree 64 Bit Multiplier for c1

Multiply the public key (Q) with the randomly chosen number k.so that the 256 bit cipher text c2 is obtained.

Cipher text C2 = (k * Q) … …. …. …. … (3)
Figure2 shows the Wallace tree 128 Bit multiplier for c2

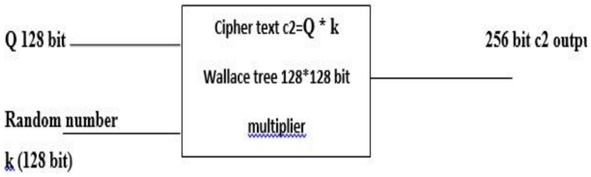


Figure 2: Wallace Tree 128 Bit Multiplier for c2

In the main generation level, by adding the message M with a point that was decided, the cipher text C2 is computed. Here 256 bit normal adder used to add M and c2, output we are getting 256 bit.
Cipher text C2 = M + (k * Q)………………………………… (4)

So at the encryption end, we have C1 & C2 to transmit to receiver. C1 and C2 will be sending
Figure3: shows the encryption block diagram of ECC Algorithm

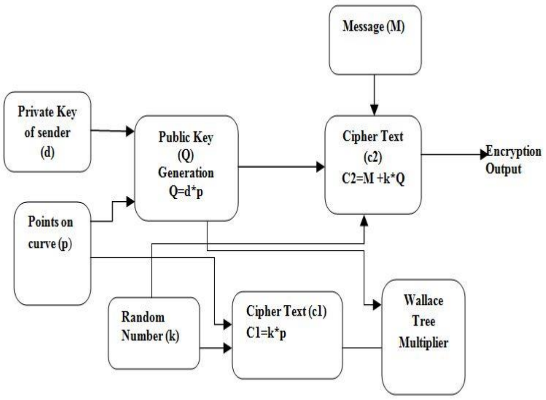


Figure 3: Encryption diagram of ECC Algorithm

DECRYPTION USING ECC ALGORITHM

On receiver, the original message may be decrypted using the same private key of sender & recipient d for decryption and using the next equation 4
[d * C1]……… (4).

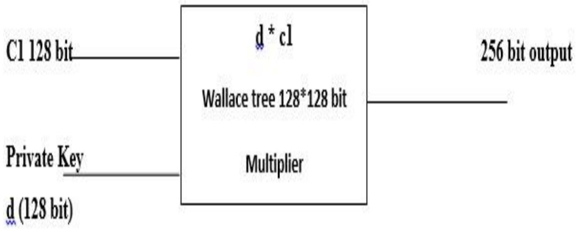


Figure 4: Wallace Tree 128 Bit Multiplier for decryption

Using subtraction, subtract the cipher text c2 from private key multiply with c1, then we obtain the original message M1.

M1 = C2 – [d * C1] … ….. (5)

And we get the original message back at the end of decryption.

Proof how does we get back the message,

- M=M1+K*Q-d*(k*P)
- M=M1+(K*d*p-d*k*p)
- M=M1 ……….ORIGINAL MESSAGE.

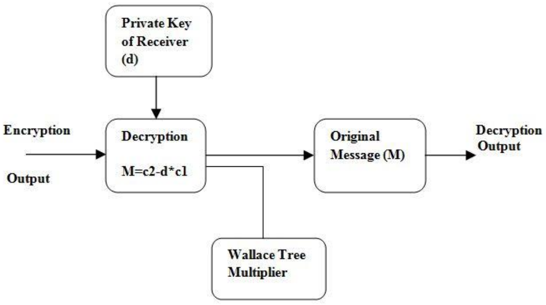


Figure 5: Decryption diagram of ECC Algorithm

RESULTS:

RTL Schematic: RTL schematic is described as register transfer logic that means the logic is transferred to registers it is also known as designer view because of it is looking like what is the intension of designer.

Figure 6 shows the RTL Schematic of Encryption

Figure 7 shows the RTL Schematic of decryption

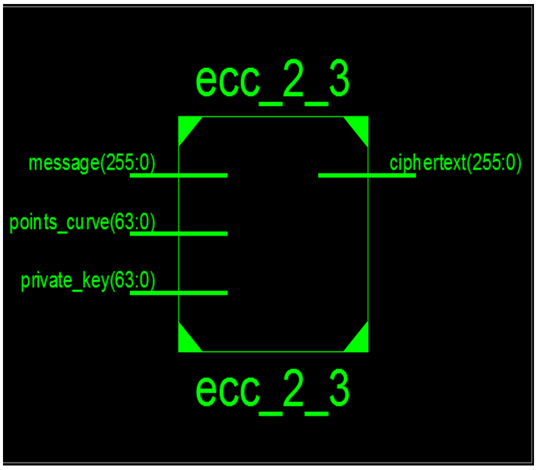


FIGURE6: RTL Schematic of Encryption

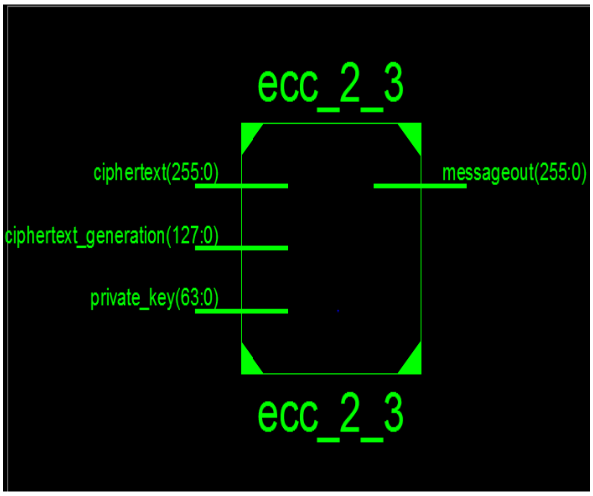


FIGURE7: RTL Schematic of Decryption

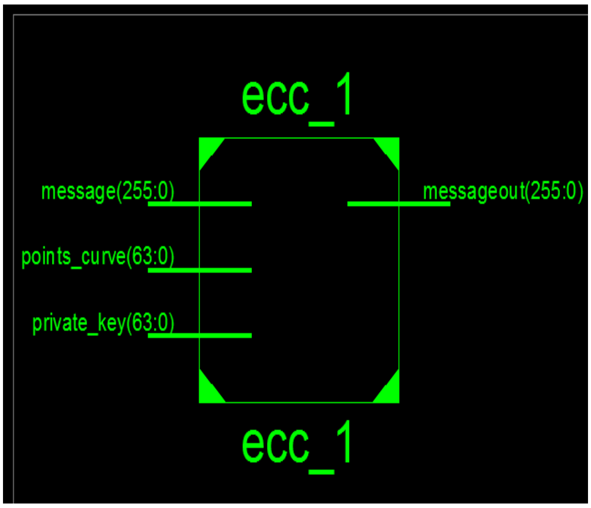
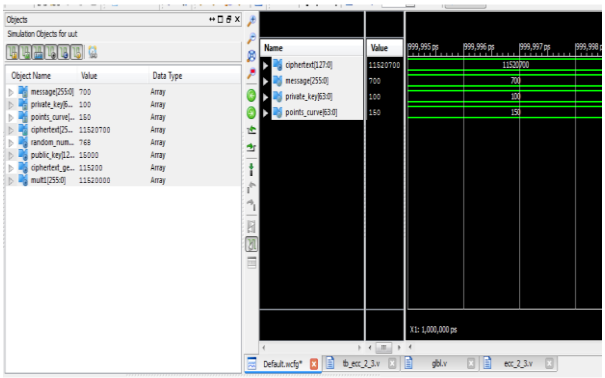


FIGURE8: RTL Schematic of ECC algorithm

SIMULATION:

The simulation is the process which is termed as the final verification in respect to its working whereas the schematic is the verification of the connections and blocks. The simulation window is launched as shifting from implementation to the simulation on the home screen of the tool, and the simulation window confines the output in the form of wave forms output. Here it has the flexibility of providing the different radix number systems.

In this project we used Verilog code written for 256 Message bit Encryption of ECC algorithm .Simulation is done by using Xilinx ISE simulator. The simulation output for Encryption of ECC Algorithm is shown below in figure 9



Objects

Simulation Objects for uut

Object Name	Value	Data Type
message[255:0]	700	Array
private_key[6...	100	Array
points_curve[...	150	Array
ciphertext[25...	11520700	Array
random_num...	768	Array
public_key[12...	15000	Array
ciphertext_ge...	115200	Array
mult1[255:0]	11520000	Array

Name	Value
ciphertext[127:0]	11520700
message[255:0]	700
private_key[63:0]	100
points_curve[63:0]	150

Fig 9: Encryption output

Inputs:

Message bit (M) = 700

Private Key (d) = 100

Points on curve (p) =150

- Random number(k)=768
- Public key(Q)=d*p=15000
- Cipher text generation(c1)=k*p=115200

Output:

Cipher text (c2) = M+ (k*Q)

= 700+ (768*15000)

c2 = 11520700

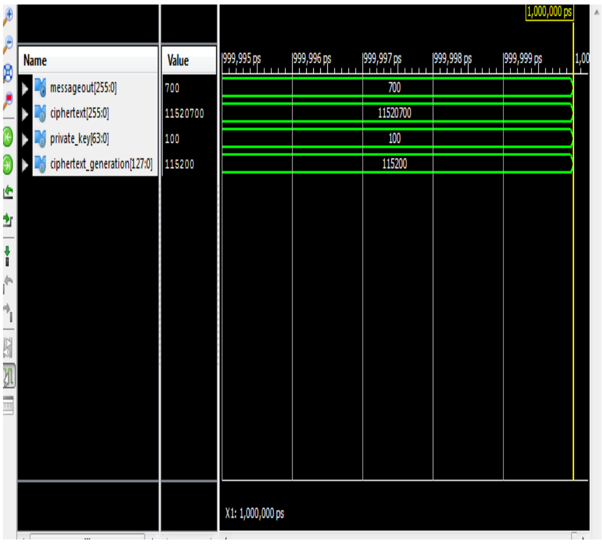
Output:

cipher text (c2) = M+ (k*Q)

= 700+ (768*15000)

c2 = 11520700

In this project Verilog code is written for Decryption of ECC algorithm .Simulation is done by using Xilinx ISE simulator. Here the input are Cipher text, Private Key and Cipher text generation and the resultant output is Message bit. So, the output of Decryption is Message bit... The simulation output for Decryption of ECC Algorithm is shown below in figure 10



Name	Value
messageout[255:0]	700
ciphertext[255:0]	11520700
private_key[63:0]	100
ciphertext_generation[127:0]	115200

Figure10: Decryption output

Inputs:

Cipher text (c2) = 11520700

Private Key (d) = 100

Cipher text generation (c1) =115200

Output

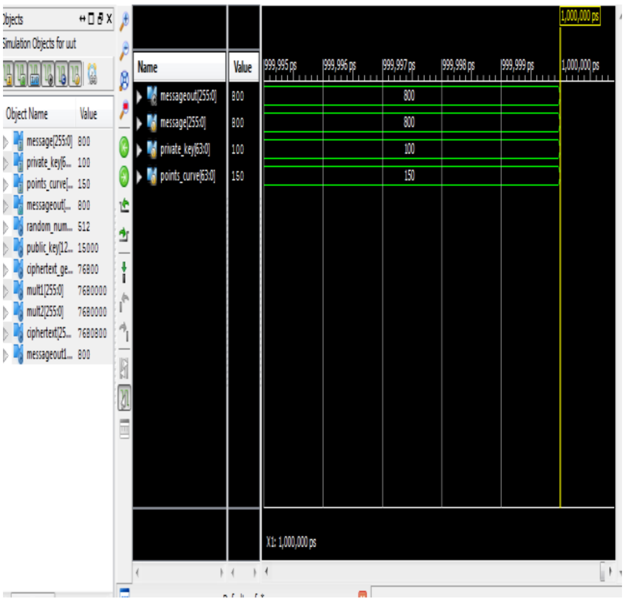
Message output (M) =c2- (d*c1)

=11520700-(100*115200)

M = 700

Here Verilog code is written for Whole ECC algorithm .Simulation is done by using Xilinx ISE simulator. Here the input is Message bit, Private Key and Points on curve and the resultant output is Message bit. So, the output of Decryption is same as given input Message bit

The simulation output for Whole ECC Algorithm is shown in figure 11



Object Name	Value
message[255:0]	800
private_key[6...	100
points_curve[...	150
messageout[...	800
random_num...	512
public_key[12...	15000
ciphertext_ge...	76800
mult1[255:0]	7680000
mult2[255:0]	7680000
ciphertext[25...	7680800
messageout1...	800

Name	Value
messageout[255:0]	800
message[255:0]	800
private_key[63:0]	100
points_curve[63:0]	150

Fig 11: ECC simulation output

Inputs:

Message bit (M) = 800

Private Key (d) = 100

Points on curve (p) =150

- Public key(Q)=d*p=15000
- random number(k)=512
- Cipher text generation(c1)=k*p=76800
- cipher text(c2) = M+(k*Q)=7680800

Output:

Message output (M) =c2-(d*c1)

=7680800-(100*76800)

M = 800

CONCLUSION & FUTURE SCOPE

CONCLUSION

In this project the area is reduced by using the Wallace tree multiplier and single point method. Previously the paper was implemented for 16_bit Vedic multiplier and in this project ECC Algorithm code is implemented for 128_bit Wallace tree multiplier. And, in view of the performance, ECC offers much faster encryption and decryption for smaller key sizes than RSA, finally the performance also increased. The ECC Algorithm is designed coded and verified through simulation and synthesis results using Xilinx tools.

FUTURESCOPE

With the short development of communication terminals and networks, customers could reap masses of offerings dispensed over the sector, on every occasion and wherever. Nonetheless, increasingly safety issues save you the advanced technology from moving forward, and increasingly more human beings start to difficulty about the safety problems of their records and conversation packages. In future, the use of Elliptic Curve Cryptography on the net of issue (IoT). Most of the device relies upon at the internet, so it's miles IoT this is a brilliant evolution of technology. Smart home, metropolis, college are some familiar examples of the net of factors. It’s far a first-rate challenge to modernize our world into the clever world from the security attitude.

REFERENCES:

[1] Prashant Abuja, Prof. Hiren Soni. “Comparative Study of Secure and Efficient Cryptography on FPGA” Journal of Emerging Technologies and Innovative Research (JETIR), February 2018, Volume 5, Issue 2.

[2] Nawari, Mustafa, et al. "Fpga based implementation of elliptic curve cryptography. “Computer Networks and Information Security (WSCNIS), 2015 World Symposium on. IEEE,

[3] Bobade, Sunil Devidas, and Vijay R. Mankar. "VLSI architecture for an area efficient Elliptic Curve Cryptographic processor for embedded systems." Industrial Instrumentation and Control (ICIC), 2015 International Conference on. IEEE, 2015.

[4] T. Wollinger, J. Guajardo, and C. Paar, "Security on FPGA: state of the art implementations and attacks, "ACM Transactions in Embedded Computing Systems, vol.3, pp.53-59, 2004.

[5] Amir Moradi, Alessandro Barenghi, Christof Paar and Timo Kasper,"On the Vulnerability of FPGA Bit stream Encryption against Power Analysis Attacks," Proceedings of the 18th ACM conference on Computer and communications, pp: 111-124, October 20 II.

[6] Anupama T, Dr. M. B. Manjunath, "Fpga implementation of elliptic curve crypto processor over gf (2'63): A Review," International Journal of Science, Engineering and Technology Research (USETR), Volume 3, Issue 5, May 2014.

[7] A.Kaleel Rahuman, Dr. G.Athisha, "Reconfigurable Architecture for Elliptic Curve Cryptography," Proceedings of the International Conference on Communication and Computational Intelligence, pp.461- 466, December- 2010.

[8] Sonia Sophia Joseph, S Prakash, "Area Efficient Implementation of GF (2m) Multipliers for Finite Fields," International Journal of Electrical, Electronics and Data Communication, Volume-2, No.5, May-2014.