International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

IJIEMR Transactions, online available on 28th nov 2021.

Link : https://ijiemr.org/downloads/Volume-10/Issue-11

DOI: 10.48047/IJIEMR/V10/I11/33

Title **DESIGN A HIGH ACCURATE ELLIPTIC CURVE ENCRYPTION AND DECRYPTION USING HOMOMORPHIC ALGORITHM**

volume 10, Issue 11, Pages: 217-224

Paper Authors: **D. NAGA JYOTHI, N. VAMSI KRISHNA, SRINIVASARAO**

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

DESIGN A HIGH ACCURATE ELLIPTIC CURVE ENCRYPTION AND DECRYPTION USING HOMOMORPHIC ALGORITHM

[1]D. NAGA JYOTHI, [2]N. VAMSI KRISHNA, [3]SRINIVASARAO
[1]M.Tech Scholar, Dept of ECE, Chebrolu Engineering College, Chebrolu, Andhra Pradesh, India
[2,3]Assistant Professor, Dept of ECE, Chebrolu Engineering College, Chebrolu, Andhra Pradesh, India

ABSTRACT: Due to privacy leakage of sensitive data, the conventional encryption systems are not completely secure from an intermediary service like cloud servers. The homomorphic encryption is a special kind of encryption mechanism that can resolve the security and privacy issues. In this project, design and implementation of high performance elliptic curve homomorphic cryptography algorithm for communication is done. Initially, input bits and key is expanded serially. Next, bits are substituted using S-Box. After that shifting and mixing operation is performed. Now these bits are encrypted. Here, a high-performance elliptic curve point multiplication is used by the efficient finite-field arithmetic unit in affine coordinates, where elliptic curve point multiplication is the key operation of an Elliptic curve based Cryptographic (ECC) processor. Similarly, decryption process is reverse to this operation. Hence elliptic curve point multiplication based homomorphic encryption and decryption is implemented and it gives better security compared to exist one. The proposed design is synthesized in field-programmable gate array (FPGA) technology with the VHDL. This system will provide better security, resource efficiency and high performance compared to existing standards. This elliptic curve based homomorphic encryption technique guarantee both privacy and integrity.

KEY WORDS: Cryptography, Homomorphic Encryption, FPGA, Elliptic Curve Based Cryptographic) ECC Processor.

I. INTRODUCTION

The confidential and private data through an internet or computer networks, there is a chance of getting threats to integrity of data, data confidentiality and availability of data because it provides the worldwide communication. The data integrity, data confmaintained by the data encryption [1].

In everyday of life, information become most important advantage in the growing of demand and it is need for storing the every single event significance. Securing of messages is necessity from unauthorized party's. To protect the information from public accesses an Encipherment is used and it is the one of the security mechanism. The original content of a message can be hiding with the help of encryption, so it cannot be readable for everyone except a person who has special ability to read it. In older days, the meaning of cryptography is, the secret keys are only used by the encryption and decryption, but nowadays cryptography can be defined in various methods such as asymmetric key encipherment it also known as a public key cryptography as well as symmetric key encipherment it also called as a private key cryptography [2]. Therefore computation time is more high in public key algorithm and it is quite complex. Moreover, a single key can be used for both decryption and encryption in the private key algorithms whereas, 2 keys are used in public key algorithm that is one key is used for encryption and another key is used for decryption [3].

An additional cryptographic algorithms are Data Encryption Standard (DES), 2-DES, 3-DES, Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC) and other algorithms [4-6]. By using side channel attacks and brute force, more number of investigators and hackers are always try to stop the cryptographic algorithms. Moreover some attacks were successful as it was the case for the Data Encryption Standard (DES). The strongest published cryptographic algorithm is Advanced Encryption Standard (AES).

Homomorphic encryption is a method of encryption that allows calculation on ciphertexts, generating an encrypted result which, when decrypted, matches the result of the operations as if it was performed on the plaintext. Homomorphic encryption [7] can be used for secure outsourced computation, such as secure cloud computing services, and securely chaining together different services without exposing sensitive data. In highly controlled industries, such as healthcare, homomorphic encryption can be used to enable new services by removing privacy blockades inhibiting data sharing. For example, analytics in health care can be hard to use due to medical data privacy concerns, but if the predictive analytics service provider can function on encrypted data in place of these privacy concerns are diminished. Homomorphic encryption schemes are inherently soft. In terms of malleability, homomorphic encryption schemes have weaker security properties than non-homomorphic schemes. A cryptosystem that supports arbitrary computation on ciphertexts [8][9] is known as fully homomorphic encryption (FHE). In this paper elliptic curve homomorphic cryptography Encryption algorithm is proposed.

II. EXISTED METHOD

Memory-based FFT, is perceived as an increasingly doable answer for low territory unpredictability, particularly for enormous size FFT (Fast Fourier Transform). A similar end can likewise be connected to NTT (Number theoretic Transform), which has similar information stream as the conventional FFT however with an alternate arrangement of twiddle factors [10]. Accordingly, memory-based FFT/NTT arrangements are appropriate to (Fully Homomorphic encryption) FHE applications that endeavor to quicken enormous whole number duplication by ASIC/FPGA structure with limited equipment cost [11]. For memory-based FFT/NTT engineering structures, proficient memory the executives plans are typically requested to build the equal memory data transmission by apportioning the required memory into a few banks. High-radix BUs are regularly connected to lessen the quantity of activity stages, in this manner expanding the subsequent presentation. There are dependably compromise amid equipment cost and time intricacy for a given application requirements.

Fig. 1 delineates the existed enormous huge integer number multiplication design, which comprises of two NTT units, a determination conveys unit, an AGU, a controller unit, and a few memory units. A NTT unit involves one radix-r BU, r 64-bit measured multipliers (MulMod), one radix-r switcher, and one support [12]. Every one of the two NTT units gets to information from two single-port SRAM banks. The ROMs are utilized to store the related twiddle factors, i.e., the forces of the crude Nth base of solidarity in Zp, for NTT/INTT calculations. The accompanying condenses the activities of the proposed NTT-based multiplication engineering [13]. The NTT1 and NTT2 units are utilized to do NTT calculations of the two information operands in the meantime. What's more, we reuse the NTT1 unit to perform INTT calculation of the consequence of point-wise multiplication. Each NTT input information has 64 bits and the radix-r BU is utilized to process r input information. The information are stuffed utilizing the operand decrease plans Fig. 1.

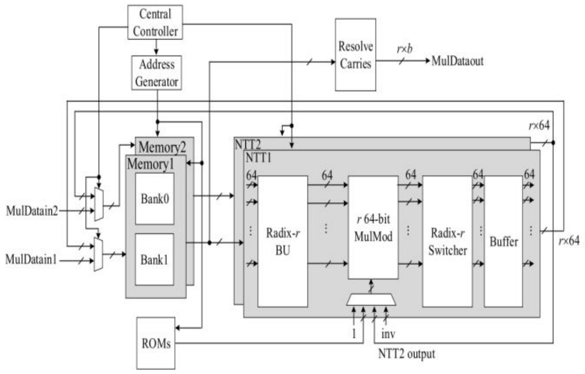

Fig. 1: FULLY HOMOMORPHIC ENCRYPTION USING LARGE INTEGER MULTIPLICATION

The MulMod unit is essentially a 64-bit modular multiplier, in which a couple of modular increases and subtractions are utilized to satisfy the multiplication of the BU yield information and a picked 64-bit esteem. The picked worth can be a twiddle factor from ROMs, a consistent estimation of 1, a BU yield information of the NTT2 unit for performing point-wise multiplication, or the opposite worth N−1 for INTT calculation, contingent upon the current operational status. The radix-r switcher is utilized to play out the transient information migration for NTT/INTT calculations [14]. Besides, to lessen the basic way delay in NTT units, the BU unit is executed in a five-arrange pipelined structure, and the MulMod unit has four pipelined stages. The buffer is utilized to store and reschedule NTT/INTT yield information for accomplishing struggle free memory get to [15]. The determination conveys unit, actualized with carry lookahead expansion, is received to deal with the conveys with the INTT yield information and get r digits of the multiplication result at once. Note that every digit of the multiplication result contains b bits for base B = 2b.

III. PROPOSED METHOD

The proposed algorithm performs operations on 64-bit plaintext and uses identical key for encryption as well as decryption. The proposed algorithm processes facts obstruct of 64-bit parts and performs 10, 12 and 14 rounds of operations employing a cipher secret of duration 64-bits, 192-bits and 256-bits respectively. The algorithm operates on data block comprised of a 4 × 4 byte matrix known as the state. The essential procedures of proposed algorithm are carried out on the state. The operations of proposed elliptic curve homomorphic cryptography Encryption algorithm with 64-bit key size are show in Fig. 2

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

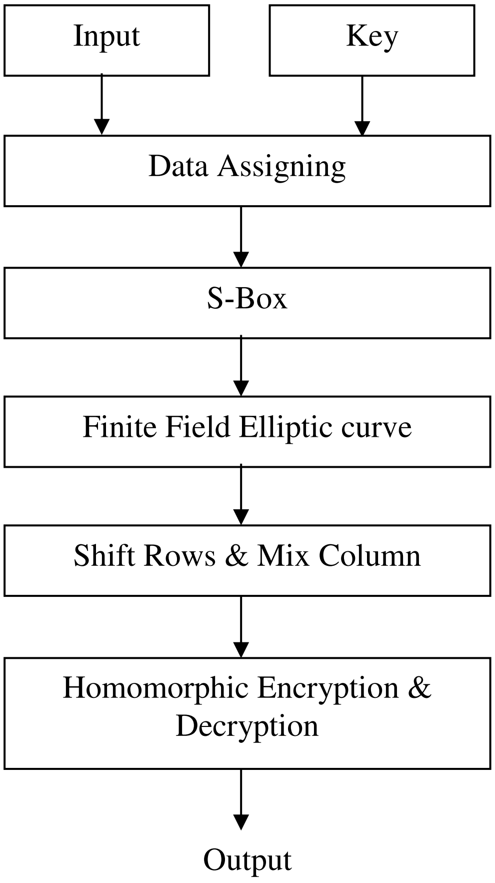
Input
Key
Data Assigning
S-Box
Finite Field Elliptic curve
Shift Rows & Mix Column
Homomorphic Encryption & Decryption
Output

Fig. 2: PROPOSED SYSTEM FRAMEWORK

In the initial round, the 64-bit plaintext is Exclusive-ORed with 64-bit initial key. In each cipher round, key expansion, S-Box, ShiftRows() and MixColumns() transformations are performed on a two dimensional 4×4 array of bytes called the states.

3.1 Data Assigning

Based on 128 – bit initial key the key expansion module generates 128 - bit keys for algorithm each round. The module of key expansion contains SubBytes, ShiftRows and RoundConst functions. Bitwise XOR operation is performed by Roundconst function utilizing round constant array. The Roundconst array consists values, that are given as $[X^{i-1}, \{00\}, \{00\}, \{00\}]$ with X^{i-1} being powers of x (x denoted as {02}) in the field GF(2^8). Hence each round key is column wise generated using

3.2 S-BOX

The S-box is formed of a lookup table of size 256 bytes. The S-box values are calculated by taking multiplicative inverse in finite field GF(2^8) where input element with all bits zero is mapped to itself and applying affine transformation over GF(2).

Expresses the affine transformation on GF(2^8).

$$S(y) = Affine\ transform\ (y^{-1})\text{-- (1)}$$

$$Affine\ transform =$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} i_7 \\ i_6 \\ i_5 \\ i_4 \\ i_3 \\ i_2 \\ i_1 \\ i_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \text{---- (2)}$$

S-box values generation using 8-bit PN Sequence Generator: A PN Sequence Generator is used to generate sequence of pseudorandom binary numbers. A PN Sequence Generator is designed using Linear Feedback Shift Register (LFSR) described by the Generator Polynomial. LFSR is shift register whose input bit is a linear function of previous state and is generated by XORing selected bits from all the bits of the shift register. The number of states generated by the LFSR is determined by the feedback taps of the generator Polynomial.

The bits are XORed and feedback from MSB side on each clock cycle resulting in cyclic shifting of previous value. Thus, a random sequence with a very large repetition period is obtained by combining elements from taps of the shift register and giving a feedback to the input of the generator. The randomness in the output values generated from PN Sequence Generator depends not only on the feedback taps but also on the non-zero initial 8-bit seed value given to the generator. The change in the seed value shifts the starting value and changes the sequence of the generated values. This results in generating sequence known only to the designers. These values can then be used to form the S-box. The invertible S-box is responsible to strengthen the AES algorithm against various attacks. The lack of knowledge of the taps and seed value selected to the attackers will make the AES algorithm invulnerable to attacks. The individual bytes from the state matrix are replaced with the corresponding value stored in modified S-box in SubBytes Transformation. For this, the higher nibble and lower nibble of the individual entry from state matrix is taken as row and column number respectively of the S-box.

3.3 Shift Rows Transformation

The ShiftRows transformation shifts rows 1, 2 and 3 of the State matrix cyclically towards left by 1, 2 and 3 positions respectively. The offset value is dependent on the row number. Thus the first row remains unchanged. Cyclic rotation of rows imparts diffusion property in algorithm. The proposed work suggests technique to modify the S-box values using PN Sequence Generator to improve the quality of encryption. The initial key required for encryption/decryption is also generated using the PN Sequence Generator instead of using a pre-defined key. In the proposed work, 8-bit PN Sequence Generator is used for generating the S-box values and initial key. The key and plaintext sensitivity tests are performed as per Strict Avalanche Criterion. The avalanche values for traditional AES algorithm are compared with avalanche values for proposed algorithm with modified S-box values. For this comparison, pre-defined initial keys as well as initial keys generated using 8-bit PN Sequence Generator are considered. Further the proposed design is synthesized using different FPGA devices and comparison with existing FPGA implementations for speed and area optimization is done.

3.4 Mixcolumns Transformation

The MixColumns transformation performs operations on each column of the state matrix one at a time. It is a linear diffusion process. Each column of the state matrix is considered as a four-term polynomial over GF(2^8).

3.5 Elliptic Curve Cryptography (ECC)

ECC is the most popular public-key encryption technique. To encrypt data in ECC, it is denoted as a point on an elliptic curve (EC) over a Galois field. A Galois field denoted normally as $GF(q = p^m)$ is said to be a binary field or characteristic-two finite field if $q = 2^m$. A elliptic curve defined over a Galois field provides a group structure that is used to implement cryptographic systems. The group operations are EC point addition (ECPA) and EC point doubling (ECPD). There are various coordinate systems to represent elliptic curve points. They vary in the number and type of field operations required to implement PA/PD. In our work, we implement all elliptic curve operations in an affine coordinate system.

3.6 Homomorphism Encryption & Decryption

A homomorphic encryption scheme is an augmented encryption scheme with two additional routines HE.Add() and HE.Mult() to perform add or multiply on encrypted data. Due to its mathematical homomorphism, the result is still an encrypted data (called ciphertext) encrypting the sum or respectively the product of the plaintexts. Users can upload their ciphertext in an untrusted cloud and still perform computations on their ciphertext without the need for decryption. Noise is used to hide the message during encryption. With every homomorphic evaluation on the ciphertext, the noise in the result-ciphertext increases. There is also a noise threshold beyond which further homomorphic evaluations would result in decryption failures. This threshold value is called the ‘depth’ of the homomorphic scheme and it is determined by the choice of parameter set (e.g, length of data structures and size of coefficients etc.). In a simplistic view, ‘depth’ of a homomorphic encryption scheme is analogous to ‘critical path’ of a circuit. An HE scheme that supports a limited number of evaluations on ciphertext is called ‘Somewhat Homomorphic Encryption (SHE).’ When an HE supports unlimited number of evaluations on ciphertext, it is called ‘Fully Homomorphic Encryption (FHE)’ scheme.

IV. RESULTS

The Xilinx design environment was used to implement and examine the developed algorithm. The FPGA architecture of proposed algorithm is shown in Fig. 3 and Fig. 4. The below Fig. 3 and Fig. 4 shows the RTL schematic and technology schematic of Proposed Elliptic curve Homomorphic cryptography algorithm. RTL schematic is the combination of inputs and outputs. Register-transfer logic deliberation is utilized in equipment portrayal dialects (HDLs) like Verilog and VHDL to make elevated level portrayals of a circuit, from which lower-level portrayals and at last genuine wiring can be determined. Structure at the RTL level is run of the mill practice in present day advanced plan.

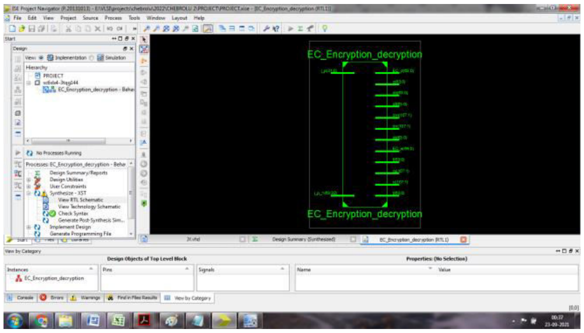

Fig. 3: RTL SCHEMATIC

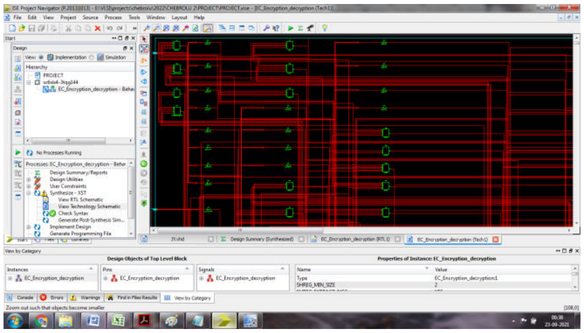

Fig. 4: TECHNOLOGY SCHEMATIC

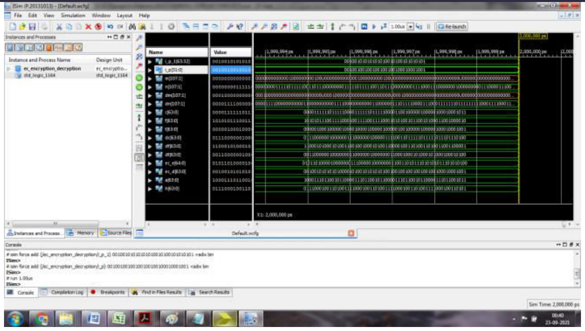

Fig. 5 OUTPUT WAVEFORMS

V. CONCLUSION

In this project, design a high-performance elliptic curve homomorphic cryptography algorithm for communication was implemented. First, the S-box values are generated by the PN Sequence Generator. Based on PN Sequence Generator the required initial key for encryption/decryption is generated. Then private key & public key can shifts the bits in one clock cycle. Depending on the homomorphic conditions the Elliptic curve based Homomorphic encryption was performed. For performing ECPD & ECPA operations an efficient polynomial – basis inversion and multiplication was developed & hence ECC processor. Within the estimated core area the EC proposed system was synthesized. The EC homomorphic cryptography was synthesized in FPGA technology with the VHDL experimental results, and this system provides security in efficient way & it is faster than CPU. The ECC proposed processor taken small amount of FPGA resources. Based on the overall performance analysis it can be concluded that this design provides better performance than others in terms of the area and the timing.

VI. REFERENCES

[1] Michela Iezzi, "Practical Privacy-Preserving Data Science With Homomorphic Encryption: An Overview", 2020 IEEE International Conference on Big Data (Big Data), 2020

[2] Rajat Sadhukhan, Debdeep Mukhopadhyay, "Design Automation for Side Channel Resistant Lightweight Cryptography", 2020 IFIP/IEEE 28th International Conference on Very Large Scale Integration (VLSI-SOC), 2020

[3] Petre Anghelescu, Ionela-Mariana Ionescu, Marian Bogdan Bodea, "Design and implementation of a visual cryptography application", 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2020

[4] Yuditha Ichsani, Resisca Audia Deyani, Rizal Broer Bahaweres, "The Cryptocurrency Simulation using Elliptic Curve Cryptography Algorithm in Mining Process from Normal, Failed, and Fake Bitcoin Transactions", 2019 7th International Conference on Cyber and IT Service Management (CITSM), Volume: 7, 2019

[5] Alshaima Q. Al-Khafaji, M. F. Al-Gailani, Hikmat N. Abdullah, "FPGA Design and Implementation of an AES Algorithm based on Iterative Looping Architecture", 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), 2019

[6] Veronica Ernita Kristianti, Eri Prasetyo Wibowo, Atit Pertiwi, Hamzah Afandi, Busono Soerowirdjo, "Finding an Efficient FPGA Implementation of the DES Algorithm to Support the", 2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT), 2018.

[7] W.-j. Lu, J.-j. Zhou and J. Sakuma, "Non-interactive and output expressive private comparison from homomorphic encryption", Proceedings of the 2018 on Asia Conference on Computer and

Communications Security ser. ASIACCS ’18, pp. 67-74, 2018

[8] Rudragoud. S. Patil, Prabhuling Biradar, “Secure Parallel Processing on Encrypted Cloud Data Using Fully Homomorphic Encryption”, 2018 4th International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2018

[9] Xiang Feng, Shuguo Li, “Design of a fast number theoretical transform engine for fully homomorphic encryption”, 2017 IEEE International Symposium on Consumer Electronics (ISCE), 2017

[10] C. Ingemarsson, P. Källström, F. Qureshi and O. Gustafsson, "Efficient FPGA mapping of pipeline SDF FFT cores", IEEE Trans. Very Large Scale Integr. Syst., vol. 25, no. 9, pp. 2486-2497, Sep. 2017.

[11] E. Öztürk, Y. Doröz, E. Savaş and B. Sunar, "A custom accelerator for homomorphic encryption applications", IEEE Trans. Comput., vol. 66, no. 1, pp. 3-16, Jan. 2017.

[12] X. Feng and S. Li, “Design of an area-effcient million-bit integer multiplier using double modulus NTT,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 9, pp. 2658–2662, Sep. 2017.

[13] Donald Donglong Chen, Gavin Xiaoxu Yao, Ray C.C. Cheung, Derek Pao, Çetin Kaya Koç, “Parameter Space for the Architecture of FFT-Based Montgomery Modular Multiplication”, IEEE Transactions on Computers, 2016

[14] X. Cao, C. Moore, M. O’Neill, E. O’Sullivan, and N. Hanley, “Optimised multiplication architectures for accelerating fully homomorphic encryption,” IEEE Trans. Comput., vol. 65, no. 9, pp. 2794–2806, Sep. 2016.

[15] W. Wang, X. Huang, N. Emmart, and C. Weems, “VLSI design of a large-number multiplier for fully homomorphic encryption,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 9, pp. 1879–1887, Sep. 2014.