DESIGN AND IMPLEMENTATION OF HIGH SECURE VLSI BASED MM-HOMOMORPHIC ENCRYPTION

[1] Dr. J. Gajendra Kumar, [2] J. Kalijappam, [3] Ch. RaghunathaBabu

[1] Professor, Department of ECE, AM Reddy Memorial College 0f Engineering and Technology, Andhra Pradesh 522601

[2] Associate Professor, Department of ECE, AM Reddy Memorial College 0f Engineering and Technology, Andhra Pradesh 522601

[3] Assistant Professor, Department of ECE, AM Reddy Memorial College 0f Engineering and Technology, Andhra Pradesh 522601

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. Unlike the public key encryption, this has three security procedures, i.e., key generation, encryption and decryption. In this project, design and implementation of high secure VLSI based MM-fully homomorphic encryption is done. This system will provide better security and resource efficiency compared to existing standards. fully homomorphic encryption and decryption technique guarantee both privacy and integrity. The main intent is to increase the speed of operation. Initially, input bits and key is given to S-Box. Next, bits are substituted using S-Box. After shifting operation is performed to the substituted bits. Now these bits are encrypted using MM homomorphic encryption. Hence MM homomorphic encryption better security compared to exist one.

KEY WORDS: Homomorphic encryption, Large Integer Multiplication, Operand Reduction, VLSI Architecture, S-Box.

I.INTRODUCTION

Fully Homomorphic Encryption is for the most part utilized in the database of the board frameworks (DMBS). One of the present issues related with the utilization of databases is the test of verifying and securely putting away the legitimate treatment of classified information in the remote database. Privacy of touchy data can be guaranteed using cryptography. It may, be the utilization of industrious encryption calculations to store the data in remote databases can fundamentally decrease the presentation of the framework without interpreting. To take care of the Issue, in MIT examines exhibited Crypto system.

Utilizing additively homomorphic crypto framework enables the server to execute SUM, AVG, and Count Questions over encoded information; the other SQL inquiries utilize the distinctive encryption calculations with the vital usefulness. The adjustment of completely homomorphic cryptosystem will keep the capacity toperform run of the mill database tasks on encoded information without decoding the information in a confided condition. In any case, such a cryptosystem must fulfill certain prerequisites for practical qualities and computational unpredictability, which is significant.

Fully Homomorphic Encryption (FHE) is a huge achievement in cryptographic research in recent years. A FHE plan can be utilized to elective perform calculations on figure content without trading off the substance of relating the plain text [1]. Therefore, a practical FHE plan will open the way to various new security advances and protection related to the applications, for example, security safeguarding pursuit and cloud-based processing. For the most part, FHE can be ordered into three classifications: cross section based, number based, and learning with mistakes.

One of the fundamental difficulties in the improvement of FHE applications is to moderate the amazingly high-computational intricacy and asset necessities [2]. For instance, programming usage of FHE in superior PCs still expend the critical calculation time, especially to achieve the vast whole number duplication which more often than not includes more than countless bits. For cross section based FHE, bit

increase the required for the little setting with a grid measurement. To quicken the FHE tasks, different effective plans have been proposed to handle the extensive whole number duplication.

The objective of this paper is to revive the encryption natives in entire number based FHE using FPGA advancement. This particular FHE count is picked because of the less unpredictable theory, humbler key size and equivalent execution. Also, the introduction of a grouped FHE plots over the entire numbers ensures further capability upgrades. Augmentation is a key segment in these FHE plans the features in the encryption, unscrambling and evaluation steps. Broad entire number FFT duplication has furthermore been used in the late of referenced gear and GPU use of other FHE plans. Future work will look into the impact of the gear multiplier on substitute walks inside the FHE plot. Specifically, presenting the primary gear execution of encryption rough required for FHE over the numbers.

ULLY homomorphic encryption (FHE) allows computations to be carried out directly on cipher texts for ensuring data privacy on untrusted servers, thus attracting much attention for cloud computing applications. Generally, FHE can be classified into three categories: lattice-based, integer based [3], and (ring) learning with errors. One of the main challenges in the development of practical FHE applications is to mitigate the extremely high-computational complexity and resource requirements. For example, software implementations of FHE in high-performance computers [4], [5] still consume significant computation time, particularly for accomplishing large integer multiplication which usually involves more than hundreds of thousands of bits. For lattice-based FHE, 785 006- bit multiplication is required for the small setting with a lattice dimension of 2048.

II. EXISTED SYSTEM

The below figure (1) shows the architecture of existed system. In this system mainly, two NTT units, a controller unit, an AGU, and several memory units are used. ROM main intent is to store the twiddle factors. There are mainly two single ports of SRAM in NTT block. Here firstly two inputs are computed at same time by using the two NTT data there are NTT1 and NTT2. For the purpose of multiplication the NTT is used as inverse NTT and because of R input data is processed.

Addition and subtraction operations are performed in the Mul Mod unit. The result of this unit is processed to the buffer unit. Now the values are saved in ROM. Here point wise multiplication process is performed in the NTT block and bits are computed depends on the current status of operation.

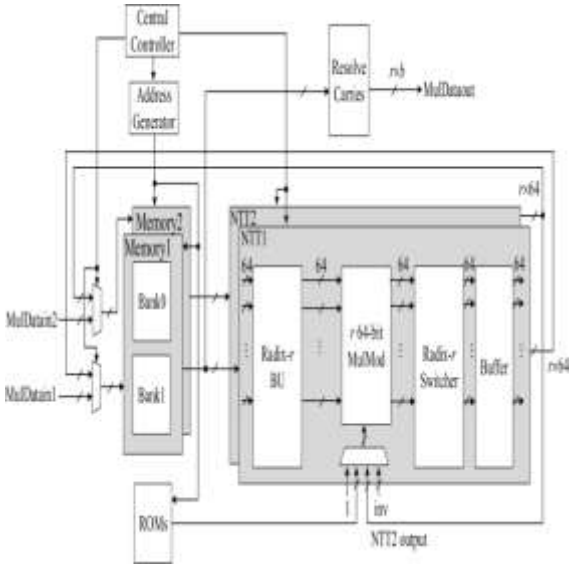

Fig. 1: EXISTED SYSTEM

To relocate the data radix r is used and this will saves the memory temporarily. Basically there are four pipelined stages in the MulMod unit. To get conflict free address in the system buffer is used. But this

system does not give effective results in terms of delay and time. Hence to overcome this, a new system is introduced which is discussed in below section.

III. PROPOSED SYSTEM

The below figure (2) shows the block diagram of proposed system. This system will provide better security and resource efficiency compared to existing standards. fully homomorphic encryption and decryption technique guarantee both privacy and integrity. The main intent is to increase the speed of operation. Initially, input bits and key is given to S-Box. Next, bits are subtitited using S-Box. After NTT is applied to the substituted bits. Now these bits are encrypted using fully homomorphic encryption. Similarly, decryption process is performed in reverse operation. The description of each block is given in detail manner.

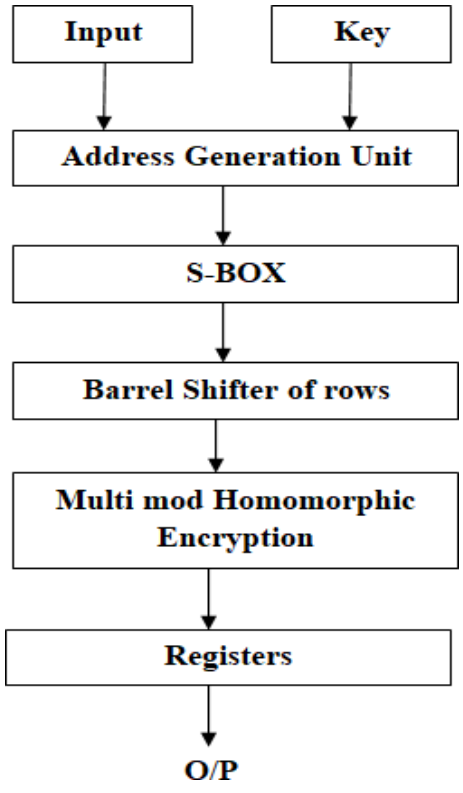

Fig. 2: PROPOSED SYSTEM

A. SUBSTITUTE BYTES TRANSFORMATION (S-BOX)

The modified structure starts with changes in the Sub bytes step. The function of this step is to substitute data present in the S-box memory unit within the state by diverse data present in other memory unit. The dispersion of data in memory units creates the confusion. The main purpose of this Shannon's contents for scientific restraint arrangement is to stimulate security. The basic purpose of substitution of bytes is to secure information.

B. ENCRYPTION

Encryption algorithm is a combination of complex mathematical functions which are used to encrypt the confidential information. Encryption key is a secret values that the sender utilizes as one of the inputs to the encryption algorithm in conjunction with plain text to generate a cipher text.

IV. RESULTS

The below figure (3) & (4) shows the RTL schematic and technology schematic of proposed system.

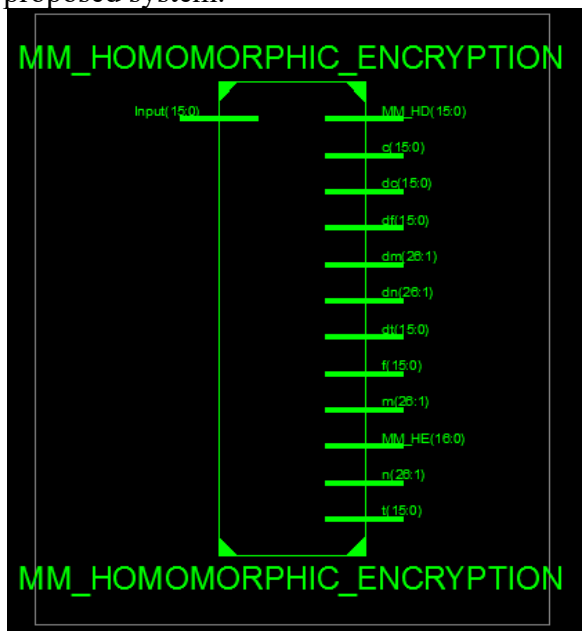

Fig. 3: RTL SCHEMATIC OF PROPOSED SYSTEM

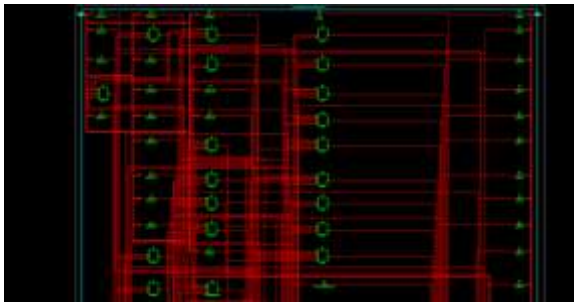

Fig. 4: TECHNOLOGY SCHEMATIC OF PROPOSED SYSTEM

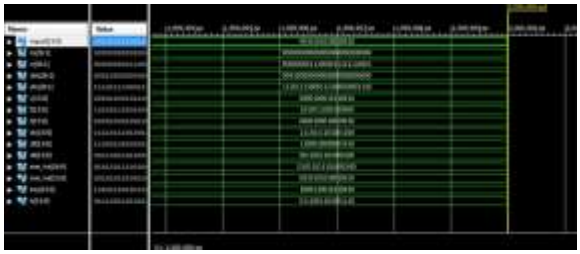

Fig. 5: OUTPUT WAVEFORM OF PROPOSED SYSTEM

V. CONCLUSION

In this project design and implementation of high secure VLSI based MM-fully homomorphic encryption was implemented. The proposed system was synthesized with an estimated core area. MM homomorphic encryption performs the operation depend on the homomorphic conditions. The public and private key will shift the bits in single clock cycle. From Experimental results it can observe that the proposed system is faster than CPU and provides security in efficient way.

VI.REFERENCES

[1] Jheng-Hao Ye and Ming-Der Shieh, "Low-Complexity VLSI Design of Large Integer Multipliers for Fully Homomorphic Encryption", 1063-8210 © 2018 IEEE.

[2] S. Koteshwara and A. Das, "Comparative study of authenticated encryption targeting lightweight IoT applications," IEEE Design Test, vol. 34, no. 4, pp. 26–33, Aug. 2017.

[3] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, and T. Unterluggauer, "ISAP–towards side-channel secure Authenticated encryption," IACR Trans. Symmetric Cryptol., vol. 2017, no. 1, pp.80–105, 2017.

[4] H. Böck, A. Zauner, S. Devlin, J. Somorovsky, and P. Jovanovic, "Nonce-disrespecting adversaries: Practical forgery attacks on GCM in TLS," in Proc. USENIX WOOT, 2016, pp. 1–11.

[5] P. G. Lopez et al., "Edge-centric computing: Vision and challenges," ACM SIGCOMM Comput. Commun. Rev., vol. 45, no. 5, pp. 37–42, Oct. 2015

[6] F. Abed, C. Forler, and S. Lucks, "General overview of the firstround CAESAR candidates for authenticated encryption," IACR Cryptol. ePrint, Tech. Rep. 2014/792, 2014.

[7] Nitesh Aggarwal, Cp Gupta, and Iti Sharma. 2014. Fully Homomorphic symmetric scheme without boot strapping. In Cloud Computing and Internet of Things (CCIOT), 2014 International Conference on.IEEE, 14–17.

[8] S Sobitha Ahila and KL Shunmuganathan. 2014. State Of Art in Homomorphic Encryption Schemes. International Journal of Engineering Research and Applications 4, 2 (2014), 37–43.

[9] D. McGrew and D. Bailey, AES-CCM Cipher Suites for Transport Layer Security (TLS), document RFC 6655, 2012.

[10] H. Handschuh and B. Preneel, "Key-recovery attacks on universal hash function based MAC algorithms," in Proc. Annu. Int. Cryptol. Conf. Berlin, Germany: Springer, 2008, pp. 144–161.