International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 2, Issue 1, pp 32-37, March-2018

Low Power Design of Highly Secured Crypto Processors

[1]Rajan Lavanya, [2]M.Vidhyia, [3]G.Kanagaraj

[1]PG Scholar, Department of VLSI, AVS Engineering College, Salem, Tamilnadu, India

[2,3]Assistant Professor, Department of ECE, AVS Engineering College, Salem, Tamilnadu, India

Abstract - In this paper we propose the design of ring learning with errors (LWE) crypto processors using Number Theoretic Transform (NTT) cores and Gaussian samplers based on the inverse transform method. The NTT cores are designed using radix-2 and radix-8 decimation-in-frequency NTT algorithms and pipeline architectures. The designed Gaussian samplers are an optimized parallel implementation of the inverse transform method and they use pipeline architecture to generate a sample every clock cycle after the latency period, that is, the output is Obtained in a fixed time achieving timing attack resistant ring–LWE crypto processors. Also, taking into account the national institute of standards and technology recommendation, a random number generator is designed to generate the input of the Gaussian sampler.

***Keywords*:** NTT, Gaussian, samplers, LWE, crypto processors.

I.INTRODUCTION

The security of the public key cryptosystems Rivest Shamir Adleman and elliptic curve cryptography relies on the hardness of factoring large integer numbers and the elliptic curve discrete logarithm problem, respectively [1]. However, the above mathematical problems can be resolved in a polynomial time using the algorithm of Short executed with quantum computing, which is an intensive research area. Then, quantum-resistant or post quantum cryptosystems have been proposed and National Institute of Standards and Technology (NIST) is going to standardize them [2], where lattice-based cryptography is a promising candidate for the post quantum era because its security proofs are based on worst case hardness of lattice problems, and there is no known quantum algorithm that efficiently solves them [3]. One of these lattice problems is the learning with errors (LWE), which is the base for some lattice-based cryptosystems. In this context, the ring-LWE cryptosystem presented is one of the most studied lattice-based cryptosystems and its hardware implementation shows that lattice-based cryptography is practical and efficient [4].

a) Need for the study

The ring-LWE encryption requires a discrete Gaussian sampler to generate errors and it should have high precision and large tail bound, such that its statistical distance from the corresponding continuous Gaussian distribution is negligible, i.e., less than 2−90. It is important to mention that the most recent error generators for lattice-based cryptography use another error distribution, which is simpler than the discrete Gaussian distribution.

However, we use the discrete Gaussian distribution for the error generation, as in the previous ring-LWE crypto processors for public key encryption [5]. Gaussian sampler was designed using a lookup table of Gaussian distributed values whose address is generated by an linear feedback shift register, and the value of the tail bound corresponds to a statistical distance larger than2−90.The Gaussian sampler implemented using the inverse transform method for a narrow Gaussian distribution that has a statistical distance of 2−22, 2−80, or 2−100, and it obtains the sample in a fixed time [6].

The Gaussian sampler designed using the Knuth–Yao method requires few area resources. However, this sampler obtains the sample in an unpredictable time, because the method uses a random walk model to perform the sampling [7]. Then, the sampler based on the inverse transform method is more secure than the one based on the Knuth–Yao method, because the first one generates the sample in a fixed time, but it demands more area resources.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 2, Issue 1, pp 32-37, March-2018

b) Objective of the study

The main contributions of this paper are as follows:

- The design of high throughput NTT-cores using the multiple-path delay commutator (MDC) architecture.
- The optimized parallel design of two fixed-time Gaussian samplers using the inverse transform method and a pipe line architecture.
- The embedded implementation of a random number generator (RNG) according to the NIST recommendation.

II.SYSTEM OVERVIEW

Block cipher is divided into three phases:

a) Key establishment phase.

b) Pseudorandom bit sequence generation phase.

c) Encryption phase.

Each of the phases is described in detail below.

a) Key Establishment phase

In this phase, a secret key is randomly selected from the key pool and exchanged between sending and receiving nodes. The key establishment phase uses an elliptic curve over prime field to generate a large key pool for node-verification purpose. An elliptic curve over prime field is an algebraic expression and is defined by the following equation: $y2(mod\ p) = x3 + Ax + B(mod\ p)$ (1) where, A and B are the coefficients and the variables x and y take the values only from the finite field within the range of prime field [8] p. Given the values of these parameters, a large number of points on the curve can be generated using basic elliptic curve operations, known as point addition and point doubling [23]. We assume that the elliptic curve parameters (i.e., prime field p, base point G (x, y), co-efficient A and B), and chaotic map parameters (i.e., m, N, μ and β) are pre distributed securely among all sensor nodes in the WSN. Now, each SN generates a list of elliptic curve points referred to as key pool by using elliptic curve operations. When a node is required to send data packets, it randomly selects a secret key (xi ,yi) from the key pool and converts it into hash code using a pre-defined hash function [9]. Then, the hash code is shared with the destination node. The destination node retrieves the shared key by matching the received code with the hash code generated for each point of its own key pool. Upon successful retrieval of the secret key, destination node verifies the legitimacy of the source node and sends an acknowledgement. This secret key (xi, yi) is used in N-logistic tent map with other parameters to generate the random bit sequence.

b) Generation of Pseudorandom Bit Sequence

This phase involves the generation of pseudorandom bit sequences using chaotic functions. The security level of a discrete chaotic map depends on the properties of the random number generation scheme such as unpredictability and unlimited period. However, most of the chaotic maps involve high precision floating point calculation to produce the sequences of random floating-point numbers which are not suitable for resource limited SNs. However, the advantage of using N-logistic tent map is that it can deal with integer parameters and thus simplifies the computation process in SNs.

c) The Encryption Process

The overall encryption process is shown as a block diagram presented in Fig. 1 where the symbols ‘M’ and ‘XO’ denote mutation and crossover operation respectively. Confusion and diffusion are two general principles that guide the design of a block cipher. Confusion is achieved by obscuring the relationship between the cipher text and the symmetric key as best as possible. On the other hand, diffusion is achieved by dissipating the redundancy of the plaintext through spreading it over the cipher text. The proposed cryptographic scheme implements three different operations: XOR, mutation, and crossover [10]. The additive cipher XOR is secure when the key stream is as long as the plaintext. On the other hand, mutation is a process of flipping one or multiple bits in a given bit string. Crossover is a process of taking two parents bit strings and producing corresponding child bit strings by interchanging selected parts of bit strings between the parents. These two genetic operations are used in genetic algorithm to generate a new population from the existing one. In our proposed encryption mechanism, the mutation and crossover genetic operations are used as tools for introducing diffusion and confusion properties in the cipher text. The mutation process is applied to create random diversity (diffusion) in the cipher text, whereas

International Research Journal of Innovations in Engineering and Technology (IRJIET)
ISSN (online): 2581-3048
Volume 2, Issue 1, pp 32-37, March-2018

the crossover operation is used to change the order of the mutated text or image data (confusion). The main benefit of using genetic operations is that they introduce relatively fair diversity in the ciphertext. Below, we describe the encryption procedure with typical examples. We first divide the pseudorandom bit sequence generated by chaotic map into 256-bit blocks, and each block is divided into two sub-blocks of 128 bits. Then, we calculate the number of 1's in each byte as well as the sum of 1's for each two consecutive bytes in the other half of the pseudorandom bit sequence. After that, we convert the plaintext into their corresponding binary codes and group them into blocks of 128-bits. This block is XOR with the first sub-block of the pseudorandom bit sequence. Then, the mutation is performed on each byte of the XORed binary codes using the total number of 1's in each byte as the starting index of the mutation process. For example, if the number of 1's in the first byte of the second sub-block of a pseudorandom bit sequence is 7, we mutate the 7th and 8th number bits in the first byte of the XORed plaintext. Thereafter, the crossover operation is executed on mutated plaintext.

A) Block Cipher Key Generation

Block cipher encryption depends on the use of a 10-bit key shared between sender and receiver. From this key, two 8-bit sub keys are produced for use in particular stages of the encryption and decryption algorithm.

B) Key Generation Unit

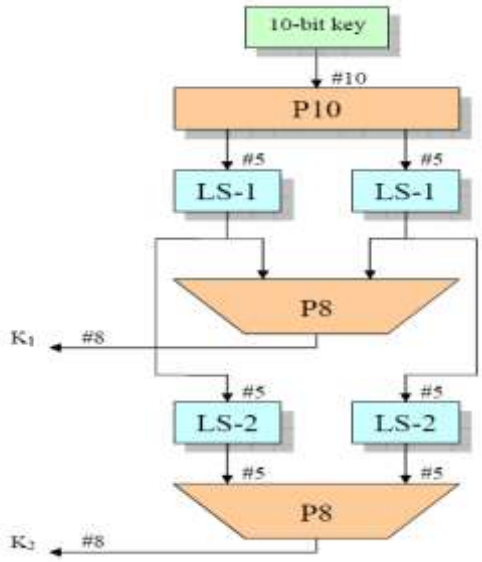

Figure-1: Key generation unit

C) Initial Permutations

- The input to the algorithm is an 8-bit block of plaintext (10111101), which we first permute using the IP function.
- **Step first:** Permute the plaintext (10111101) with IP and the value becomes (01111110).Divide the value into two parts. The left part becomes (0111) and the right part becomes(1110).
  - 

- The proposed cryptographic scheme implements three different operations: XOR, mutation, and crossover.
- The additive cipher XOR is secure when the key stream is as long as the plaintext. On the other hand, mutation is a process of flipping one or multiple bits in a given bit string.

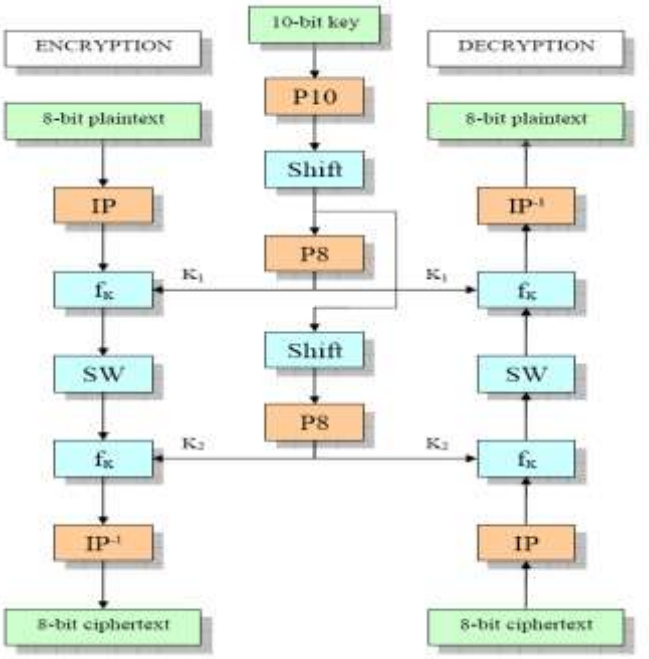

Figure-2: First stage Block cipher Encryption Round

- Crossover is a process of taking two parents bit strings and producing corresponding child bit strings by interchanging selected parts of bit strings between the parents. These two genetic operations are used in genetic algorithm to generate a new population from the existing one.

ISSN (online): 2581-3048
Volume 2, Issue 1, pp 32-37, March-2018

- In our proposed encryption mechanism, the mutation and crossover genetic operations are used to generate cipher text.
- The mutation process is applied to create random diversity (diffusion) in the cipher text, whereas the crossover operation is used to change the order of the mutated text or image data (confusion). The main benefit of using genetic operations is that they introduce relatively fair diversity in the cipher text.

III. SOFTWARE SPECIFICATION

a) XILINX FPGA Navigator 9.2I for Synthesis

The Spartan-3 generation of FPGAs includes the Extended Spartan-3A family (Spartan-3A, Spartan-3AN, and Spartan-3A DSP platforms), along with the earlier Spartan-3 and Spartan-3E families. These families of Field Programmable Gate Arrays (FPGAs) are specifically designed to meet the needs of high volume, cost-sensitive electronic applications, such as consumer products. The Spartan-3 generation includes 25 devices offering densities ranging from 50,000 to 5 million system gates, as shown in Table 1-5 through Table 1-7. The Spartan-3 platform was the industry's first 90 nm FPGA, delivering more functionality and bandwidth per dollar than was previously possible, setting new standards in the programmable logic industry.

b) Spartan-3 Generation Families

Extended Spartan-3A family

- Lowest total cost.
- Spartan-3A Platform - Ideal for bridging, differential signaling, and memory interfacing.
- Spartan-3A DSP Platform - Higher density option in Extended Spartan-3A family- DSP48A resources for digital signal processing (DSP) applications.
- Spartan-3AN Platform - Non-volatile - Ideal for space-constrained applications Spartan-3E family Spartan-3 family.

c) Architectural Overview

The Spartan-3 generation architecture consists of five fundamental programmable functional elements: • Configurable Logic Blocks (CLBs) contain flexible Look-Up Tables (LUTs) that implement logic plus storage elements used as flip-flops or latches. CLBs perform a wide variety of logical functions as well as store data.

- Input/output Blocks (IOBs) control the flow of data between the I/O pins and the internal logic of the device. IOBs support bidirectional data flow plus 3-state operation. Supports a variety of signal standards, including several high-performance differential standards. Double Data-Rate (DDR) registers are included.
- Block RAM provides data storage in the form of 18-Kbit dual-port blocks.
- Multiplier Blocks accept two 18-bit binary numbers as inputs and calculate the product. The Spartan-3A DSP platform includes special DSP multiply-accumulate blocks.
- Digital Clock Manager (DCM) Blocks provide self-calibrating, fully digital solutions for distributing, delaying, multiplying, dividing, and phase-shifting clock signals. These elements are organized as shown in Figure 1-1, using the Spartan-3A FPGA array as an example. A dual ring of staggered IOBs surrounds a regular array of CLBs in the Spartan-3 and Extended Spartan-3A family. The Spartan-3E family has a single ring of inline IOBs. Each block RAM column consists of several 18-Kbit RAM blocks. Each block RAM is associated with a dedicated multiplier. The DCMs are positioned with two at the top and two at the bottom of the device, plus additional DCMs on the sides for the larger devices. The Spartan-3 generation features a rich network of traces that interconnect all five functional elements, transmitting signals among them. Each functional element has an associated switch matrix that permits multiple connections to the routing.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 2, Issue 1, pp 32-37, March-2018

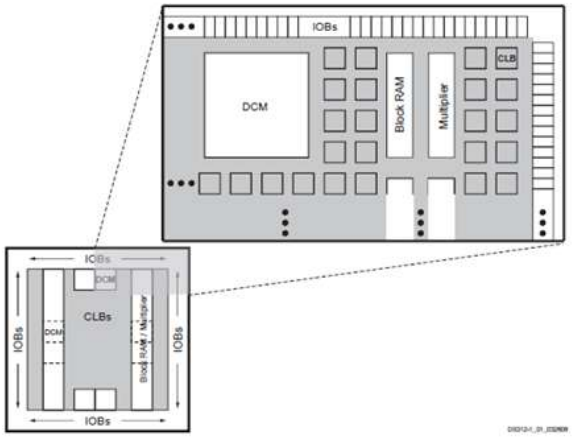

Figure-3: Spartan-3A Platform Architecture

IV. SIMULATION RESULT

This is the simulation result from Xilinx The design of ring learning with errors (LWE) crypto processors using Number Theoretic Transform (NTT) cores and Gaussian samplers based on the inverse transform method.

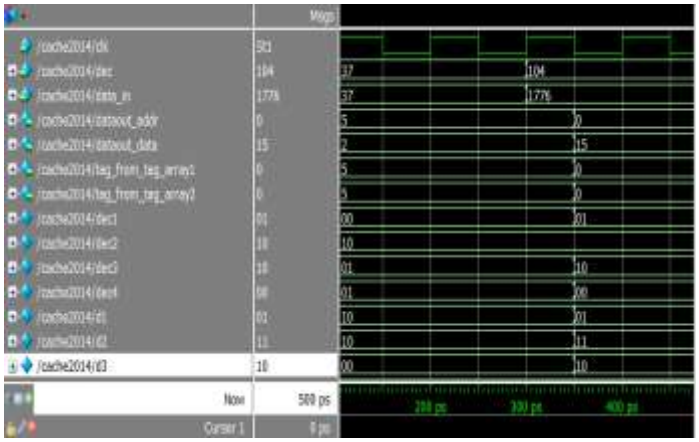

Figure-4: Simulation Result

The NTT cores are designed using radix-2 and radix-8 decimation-in-frequency NTT algorithms and pipeline architectures.

V. CONCLUSION

The design of ring learning with errors (LWE) crypto processors using Number Theoretic Transform (NTT) cores and Gaussian samplers based on the inverse transform method. The NTT cores are designed using radix-2 and radix-8 decimation-in-frequency NTT algorithms and pipeline architectures. The designed Gaussian samplers are an optimized parallel implementation of the inverse transform method and they use pipeline architecture to generate a sample every clock cycle after the latency period, that is, the output is Obtained in a fixed time achieving timing attack resistant ring–LWE crypto processors. Also, taking into account the national institute of standards and technology recommendation, a random number generator is designed to generate the input of the Gaussian sampler.

REFERENCES

[1] Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," *J. Cryptol.*, vol. 18, no. 4, pp. 291–311, 2005.

[2] Bogdanov, D. Khovratovich, and C. Rechberger, "Biclique cryptanalysis of the full AES," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 7073. Berlin, Germany: Springer-Verlag, 2011, pp. 344–371.

[3] De Cannière, O. Dunkelman, and M. Kneževi´c, "KATAN and KTANTAN—A family of small and efficient hardware-oriented block ciphers," in *Cryptographic Hardware and Embedded Systems*, vol. 574 . Berlin, Germany: Springer-Verlag, 2009, pp. 272–288.

[4] *Intel Architecture Software Developer's Manual*, Intel Corporation, Santa Clara, CA, USA, 1997.

[5] J. Black, P. Rogaway, and T. Shrimpton, "Encryption-scheme security in the presence of key dependent messages" in *Selected Areas in Cryptography* (Lecture Notes in Computer Science), vol. 2595. Berlin, Germany: Springer-Verlag, 2003, pp. 62–75.

[6] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science). Berlin, Germany: Springer-Verlag, 2011, pp. 326–341.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," *ACM SIGPLAN Notices*, vol. 35, no. 11, pp. 93–104, 2000.

ISSN (online): 2581-3048
Volume 2, Issue 1, pp 32-37, March-2018

[8] J. Kumar and S. Nirmala, "Encryption of images based on genetic algorithm—A new approach," in *Advances in Computer Science, Engineering & Applications* (Advances in Intelligent Systems and Computing), vol. 167. Berlin, Germany: Springer-Verlag, 2012, pp. 783–791.

[9] K. Biswas, V. Muthukkumarasamy, E. Sithirasenan, and K. Singh, "A simple lightweight encryption scheme for wireless sensor networks," in *Distributed Computing and Networking*,vol. 8314. Berlin, Germany: Springer-Verlag, 2014, pp. 499–504.

[10] M. Amara and A. Siad, "Elliptic curve cryptography and its applications, in *Proc. 7th Int. WOSSPA*, May 2011, pp. 247–250.

How to cite this article:

Rajan Lavanya, M.Vidhyia, G.Kanagaraj, "Low Power Design of Highly Secured Crypto Processors", in *International Research Journal of Innovations in Engineering and Technology (IRJIET)*, Volume 2, Issue 1, pp 32-37, March 2018.

****

