International Journal of Advanced Science and Technology
Vol. 29, No. 12s, (2020), pp. 2257-2277

Hardware Efficient Hybrid Wireless Crypto Processor Using Enhanced Advance And Side-Channel Resistant Authenticated Encryption Standard

Abstract

In this modern era, communication or data transmission plays an important role in a human's life and carried out in wireless medium. Cryptography techniques are necessary for confidential data transmission in wireless media, which protects electronic data in communication network. Many algorithms that are cryptographically secure are not easily implemented in computer applications especially in hardware. In this paper, we propose a hardware efficient hybrid wireless crypto processor (HWCP), which combines two block ciphers such as enhanced advanced encryption standard (AES) and side-channel resistant authenticated encryption with masking (SCREAM). Generally, the hardware cost of hybrid processors are very high, here we use composite field arithmetic (CFA), on the fly key expansion, and order change to reduce the hardware parts in the encryption algorithms. The main objective of this design is to propose hybrid crypto processor, which is complex and not easy to crack the keys from malicious. The proposed HWCP design maximizes the security via increasing the complexity of cracking keys. Moreover, the proposed HWCP design is implemented with parallel sub-pipeline manner that increases the throughput. The proposed HWCP design synthesized with different FPGA families are Virtex-6 (xc6vlx75t-2), Artix-7 (xc7a100t-2), Kintex-7 (xc7k70t-2) and Virtex-7 (7vx330t-2) in Xilinx tool. The performance comparison of proposed HWCP design is compared with existing designs in terms of hardware utilization, power consumption and maximum operating frequency.

Keywords: *hybrid cryptography, block ciphers, AES, SCREAM, field programmable gate array*

1. Introduction

International telecommunications union (ITU) in 2013 reveals that there are more than 6 billion subscribers worldwide and more than 40% of the world's population have access to the internet [1]. It predicts the wireless medium plays major role in data communication and the survivability relates to security, robustness against attacks and failure of wireless communication itself [2]. In order to achieve data confidentiality, data authentication and replay protection, authenticated encryption should be used [3]. Cryptography algorithm is the solution for this problem, which is the art of protecting data by transforming and technology application. It provides number of security goals to ensure of privacy of data, on-alteration of data [4]. The idea of encryption and encryption algorithm used to encode data in secret code and not to be able readable by hacker's unauthorized person even it is hacked. In order to perform encryption and decryption the various cryptographic techniques are used such as advanced encryption standard (AES) [5], data encryption standard (DES) [6], triple DES [7], Blowfish [8], Rivest-Shamir-Adleman (RSA) [9], ElGamal [10] and Paillier [11]. These techniques have different key size, block size and number of round and each method has different execution time and throughput. Due to the security issues in modern communication system the cryptographic techniques that have as vulnerabilities as possible while maintaining their low implementation complexity. The mentioned algorithms [6]-[11] are functionally secure but it not easier to implement in computer applications like hardware.

Copyright ⓒ 2020 SERSC

Thus, the need for hardware implementations of secure algorithms becomes even greater for security enhancement [12].

Various hardware cryptosystems have been proposed in which the choice of hardware may be microcontrollers, microprocessors, and custom application-specific integrated circuit (ASICs) based cryptosystems [13]. Each of the aforementioned hardware offer some merits and demerits, for instance, a microcontroller based design might have low processing capability but such a design usually takes low time to market. Similarly, an ASIC based solution can achieve very high data rates and power efficiency but require high time to market. Hardware based solutions with high performance and low power can be designed on custom ASIC platform. ASIC designs are usually produced in mass volumes, so their per unit cost is also low but these solutions have high time to market as the generation of ASIC designs is a very complex process and in case of any error in the design solution is redesigned which increases the non recurring engineering (NRE) cost. For a high performance solution with low cost and low power consumption, FPGA based design is another candidate [14]. These designs have very low time to market and have very low NRE cost of FPGAs due to the re-configurability. The speed and efficiency of FPGAs combined with their flexibility makes them very attractive for cryptographic applications. The ability to design an FPGA to use a different cryptographic algorithm on the fly or to be able to update, modify outdated algorithm make them very useful for designing cryptosystems. A highly parallelized RC4 [15] key searching system is used for an effective hardware implementation of a parallel key searching system in a Xilinx XC2VP20-5 chip. Energy efficient and security-optimized hardware AES module [16] is implemented based on the combination of twelve DSE and four Galois field (GF) S-Boxes, and the latch dividing data path. C-testable S-box implementation is one of the most complex blocks in AES hardware implementation [17]. Then, divide the S-box structure into a positive polarity Reed–Muller form and tested independently using a BIST circuit. FPGA-based secure hash algorithm (SHA-1) cryptanalysis system [18] is used for a case-study achieved an elementary operation collision much higher than others. A charge-sharing symmetric adiabatic logic (CSSAL) [19] in an 8-bit S-box circuit is implemented in using a multi-stage positive polarity Reed–Muller representation with a composite field technique. A 64 bit block ciphers MISTY1 and KASUMI [20] is comprised of reconfigurable components consisting of FL function, first out/in (FO/FI) function and XOR function designed to perform MISTY1 and KASUMI algorithms round transformation functions.

Contributions of this paper: An efficient hybrid wireless crypto processor (HWCP) is proposed, which combines the efficient two block ciphers are advanced encryption standard (AES) and side-channel resistant authenticated encryption masking (SCREAM). The hardware complexities of both the crypto techniques are enhanced by the composite field arithmetic (CFA), on the fly key expansion, and order change to reduce the hardware utilizations. The main objective of proposed HWCP design maximizes the security via increasing the complexity of cracking keys.

The remainder of this paper is organized as follows. In Section 2, we review the basics of hybrid crypto techniques for data communications. In Section 3, we present the proposed methodology and system model of proposed HWCP design; the detailed working function and their mathematical models are described in Section 4. The simulation results and the performance analysis are presented in Section 5. Finally, the paper concludes in Section 6.

2. Related works

Copyright ⓒ 2020 SERSC

Liu et al. [21] have investigated the elementary operations in AES and analyzed the logic depth of each operation. This is because a great logic depth will increase the signal delay, and thus reduce the maximum clock frequency of the AES hardware implementation. The logic depth of the operations to FPGA logical architectures in mathematical formulation and it reveals the critical path of the AES hardware design, and provide guidance on pipelining designs. With the aid of the analytical formulae by two pipelining solutions and combining combinational logic of the elementary operations in single AES cipher round into stages, with the goal of achieving balanced and reduced logic depth. The elementary operation level divides the single AES cipher round into two pipelining stages. This design is implemented on the Virtex7 XC7VX690T, XC7VX690T and consumes 4339 slices, 593.12 MHz maximum frequency and 38511 slices, 454.55 MHz maximum frequency respectively.

Kundi et al. [22] have proposed a unified FPGA based AES encryption/decryption design for a symmetric ST-Box structure. This structure fully utilizes high capacity block RAM (BRAM) by accommodating all encryption and decryption lookup operations within a single BRAM in the form of single integrated look-up-table (LUT). This design also caters the inherent asymmetric nature of encryption and decryption coefficients for a unified hardware. The symmetry at BRAM output is maintained to use a single XOR network during both encryption and decryption. The performance of this design is enhanced by proposing a duty-cycle based accessing technique. This design is implemented on the Virtex7 FPGA family and consumes 1086 slices, 357 MHz maximum frequency and 252.89mW power consumption.

Constantin et al. [23] have presented a fast and flexible distillation engine for Quantum key distribution (QKD) and it requires only one single medium-sized industry standard FPGA per user. This distillation engine performs the complete QKD operation including synchronization, continuous real-time distillation and authentication for secret key rates up to 4 Mbps. This post-processing block size after error correction is 106 bits, and all classical communication channels are time-multiplexed in one optical channel in each direction next to the quantum channel to run the complete system continuously. This design is implemented on the Virtex-6 LX240T FPGA target device and consumes 63302 LUTs and 125 MHz maximum clock frequency.

Subramanian et al. [24] have proposed an efficient error detection architectures block ciphers LED and HIGHT including variants of re-computing with encoded operands and signature-based schemes to detect both transient and permanent faults. Authenticated encryption is applied in cryptography to provide confidentiality, integrity, and authenticity simultaneously to the message sent in a communication channel. This scheme is applicable to case study of simple lightweight CFB (SILC) for providing authenticated encryption with associated data (AEAD). This design is implemented on the Virtex-7 7VX330tffg1157-3 target device and consumes 217 slices, 3.67mW power for LED design and consumes 252 slices, 2.17mW power for HIGHT design.

Akleylek et al. [25] have proposed efficient modular polynomial multiplication methods with applications in lattice-based cryptography. A sparse polynomial multiplication used for quotient ring based on the modification in algorithm with sliding window method for sparse polynomial multiplication. It is used for cryptographic applications which require the sparse polynomial multiplication by a slight modification in reduction operation. This method is independent of choice of reduction polynomial and implemented on Core i5-3210M CPU platform compares with number theoretic transform multiplication.

Copyright ⓒ 2020 SERSC

Wang et al. [26] have presented reconfigurable encryption processor core (REPROC) using the interconnection tree between rows (ICTR) method that effectively reduces the complexity of the interconnections based on the characteristics of the symmetric ciphers. This technique ensures that the ratio of the interconnection area to the total area remains approximately constant, i.e., it does not increase with an increase in the array scale. A hierarchical context organization (HCO) algorithm is used for the contexts are divided into three levels are higher level contexts can call lower level contexts. This avoids duplication of contexts, such that it reduces the total size of the contexts and results in a fast and dynamic configuration. REPROC design is implemented on the Altera target device and consumes 51.35 mm^2 area, 400MHz clock frequency and 0.584W power consumption.

GranadoCriado et al. [27] have presents two hardware coprocessors are AES and international data encryption (IDES) algorithm for high-performance symmetric cryptographic algorithms. The work implementation of these algorithms is based on the two different hardware coprocessors, a FPGA and a graphics processing unit (GPU). These two devices allow implementing very fast versions of both cryptographic algorithms employing two different parallelism methodologies: The on-the-fly technique combined with this design to increase of resources and, in some cases, of the clock cycle. The pipelining is very difficult to achieve using this on-the-fly technique, because all sub-keys are needed at the same time. This design is implemented on the Virtex-2 6000 FPGA target device and consumes 3720 slices and 24.922Gbps clock frequency.

Kundi et al. [28] have presented two low-power secure hash algorithm-3 (SHA-3) designs on FPGA using embedded digital signal processing (DSP48E) for area constrained environments and the other for high-speed applications. The seven equations of SHA-3 are logically optimized to three/four stage pipelined organizations for compact and high-speed designs. The maximum parallelism between all the bitwise operations of different stages of SHA-3 is explored with respect to the 48-bit structure of DSP slice. The logical cascade structure (LCS) design strategy is used in accordance with the DSP slice organization. SHA-3 design is implemented on the Virtex-6 FPGA target device and consumes 208 slices, 130mW power consumption and 451.26MHz clock frequency.

Farooq et al. [29] have presented five different implementation techniques for AES algorithm. The use of optimizations like loop unrolling that introduced parallelism in our design. The different FPGA resource mappings have lead to different results. It is seen that generally sequential implementations lead to better area results but poor performance results. It was observed that parallelism leads to better delay results but poor area results as more registers are required. Moreover, it was observed that efficient usage of computing resources of FPGAs leads to better performance results and frequency as high as 886.64MHz can be achieved in certain scenario. AES design is implemented on the Virtex5 target device and it consumes 9276 slice LUTs, 161 LUT-FFs pair, 255 slice registers and 644.33 MHz maximum clock frequency.

Vollala et al. [30] have proposed enhanced Montgomery multiplication takes only n + 1 number of clock cycles to evaluate modular multiplication for n-bit modulus. It involves a maximum of two 1-bit additions and a right shift operation for each bit of the modulus N iterated for loop. It alleviates final subtraction for adjustment of residue in Montgomery multiplication. Montgomery multiplication is used for other cryptographic techniques are Rabin key cryptography, and

Copyright ⓒ 2020 SERSC

ElGamal scheme apart from Rivest–Shamir–Adleman (RSA). Square and multiply technique has also been tuned according to the requirements of enhanced Montgomery multiplication.

3. Problem methodology and system model

3.1 Problem methodology

At et al. [31] have described the design of a compact 8-bit coprocessor for the AES and the cryptographic hash function Grøstl. The arithmetic and logic unit (ALU) has only one instruction that allows for implementing AES encryption, AES decryption, AES key expansion, and Grøstl at all levels of security for multiple levels are 128, 192 and 256-bit encryption keys with the combination of 256 and 512 bit message digests. The key expansion algorithms of the AES generated by the continuous clock cycles and the read two bytes and write one byte at each clock cycle to avoid pipeline stalls. It suffices to store a copy of the round keys in the register file. Grøstl-n does not overwrite the round keys of the AES. As long as the AES master key is not modified, it is therefore possible to switch between the hash function and the block cipher with no need for the AES key expansion step. AES-Grøstl design is implemented on a Virtex-6 FPGA requires 169 slices and a single 36k memory block, and achieves competitive throughput up to 217Mbps and 92Mbps for encryption and hashing, respectively. A hardware efficient hybrid wireless crypto processor (HWCP) is proposed which combines two block ciphers such as enhanced advanced encryption standard (AES) and side-channel resistant authenticated encryption with masking (SCREAM). The main contributions of proposed HWCP are summarized as follows:

- Generally, the hardware cost of hybrid processors are very high, here we use composite field arithmetic (CFA), on the fly key expansion, and order change to reduce the hardware parts in the encryption algorithms.
- The main objective of this design is to propose hybrid crypto processor, which is complex and not easy to crack the keys from malicious.
- The proposed HWCP design maximizes the security via increasing the complexity of cracking keys.
- HWCP design is implemented with parallel sub-pipeline manner that increases the throughput. The design synthesized with different FPGA families is Virtex-6 (xc6vlx75t-2), Artix-7 (xc7a100t-2), Kintex-7 (xc7k70t-2) and Virtex-7 (7vx330t-2) in Xilinx tool.
- The performance comparison of proposed HWCP design is compared with existing designs in terms of hardware utilization, power consumption and maximum operating frequency.

The hardware structure of hybrid AES-SCREAM cryptosystem combines the massive structure such as parallel and sub-pipeline structure, named as hybrid parallel and sub-pipeline. Generally hybrid systems consume more area and reduce throughput of the system. The sub-pipeline concept is the best solution for reduce the area by handling more inputs with reducing the intermediate registers. The parallel architecture maximizes the throughput by handling the data in high speed by splitting modules into small number of modules and performs operations in parallel. The both parallel and sub-pipeline concepts are combine to reduce our hybrid problems. The proposed hybrid parallel and sub-pipeline architecture of AES-SCREAM is shown in fig. 1.

Copyright ⓒ 2020 SERSC

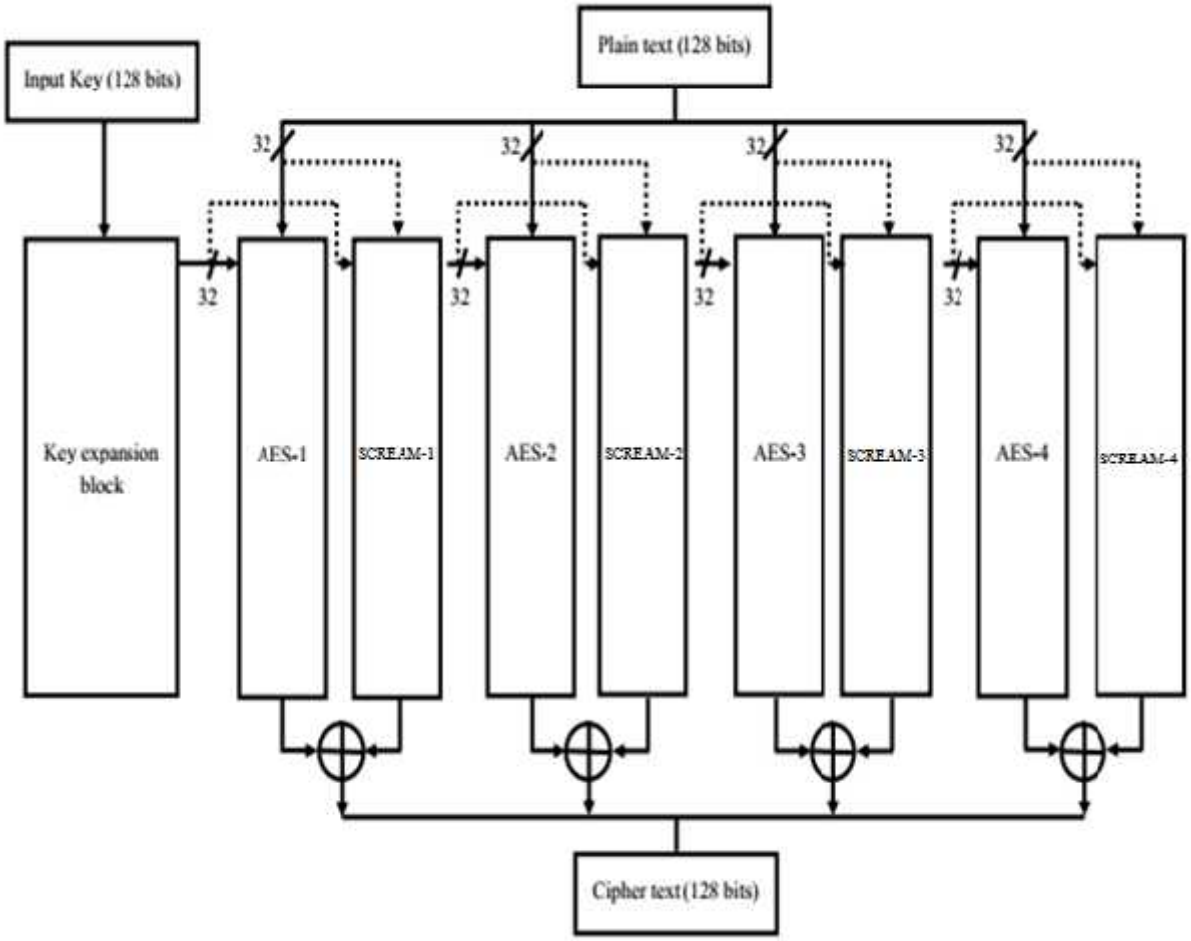


Fig. 1 Hybrid parallel and sub-pipeline architecture of AES-SCREAM

In the proposed technique, the input plain text 128 bits is divided into four 32 bits and is given to separate hardware components for both AES block (AES1, AES2, AES3 and AES4) and SCREAM block (SCREAM-1, SCREAM-2, SCREAM-3 and SCREAM-4) and thus achieving the parallel architecture. The subsections of AES rounds consist of add round key, sub bytes, shift rows and mix column with 10 rounds, and the key stream generation of SCREAM, which achieves the sub-pipelined architecture. The computed cipher text (AES_1-4 and SCREAM _1-4) is XOR with each other to form the final cipher text.

4. Hardware efficient hybrid wireless crypto processor (HWCP)

4.1 Advanced encryption standard with modified key expansion block

LUT based key expansion approach consumes more amount of area for maintaining generated keys in registers. In order to reduce the area, the keys are computed on the fly and the circuit to generated the on the fly key is shown in the fig. 2. The 128 bit input key is divided into 4 bytes, such as K0, K1, K2 and K3. The constant register (R_c)consists of the round constant values to be

Copyright ⓒ 2020 SERSC

used for each round. Initially, the K3 is rotated and the bytes are substituted from the S-Box. This value is XOR with the round constant values $(R_c[i])$ in R_c and is XOR with the K0. The resulting word and the K1 is XOR and the resulting word will be the new generated K2 which will be sent as the input for the next round input. Similarly the operation follows for the consecutive words. Again, the new generated key is fed as input to the key generation unit for the next set of key for the next round. Thus the process of generating key continues until it reaches the round 10 for AES-RC4 128 bits. The data flow graph of 128 bit on the fly key expansion is shown in Fig. 2.

The detailed hardware structure implemented from the following equations,

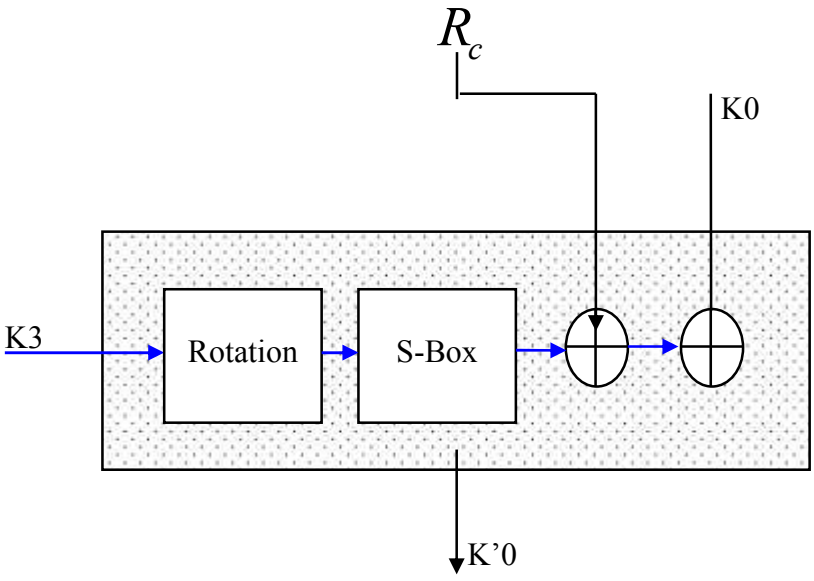
$$K'0 = Sbox\left(Rot(K3)\right) \oplus R_c[i] \oplus K0 \tag{1}$$

$$K'1 = K'0 \oplus K1 \tag{2}$$

$$K'2 = K'1 \oplus K2 \tag{3}$$

$$K'3 = K'2 \oplus K3 \tag{4}$$

The sub bytes (S-box in encryption) and the inverse sub bytes (inverse S-box in decryption) transformations are the most resource consuming operations in the steps of AES. In recent years, AES based cryptosystems are used for many applications, such as RFID tags, mobile networks (MANETs), and wireless sensor networks (WSNs). Hence CFA is employed in the sub bytes and inverse sub bytes steps to reduce the area.



Copyright ⓒ 2020 SERSC

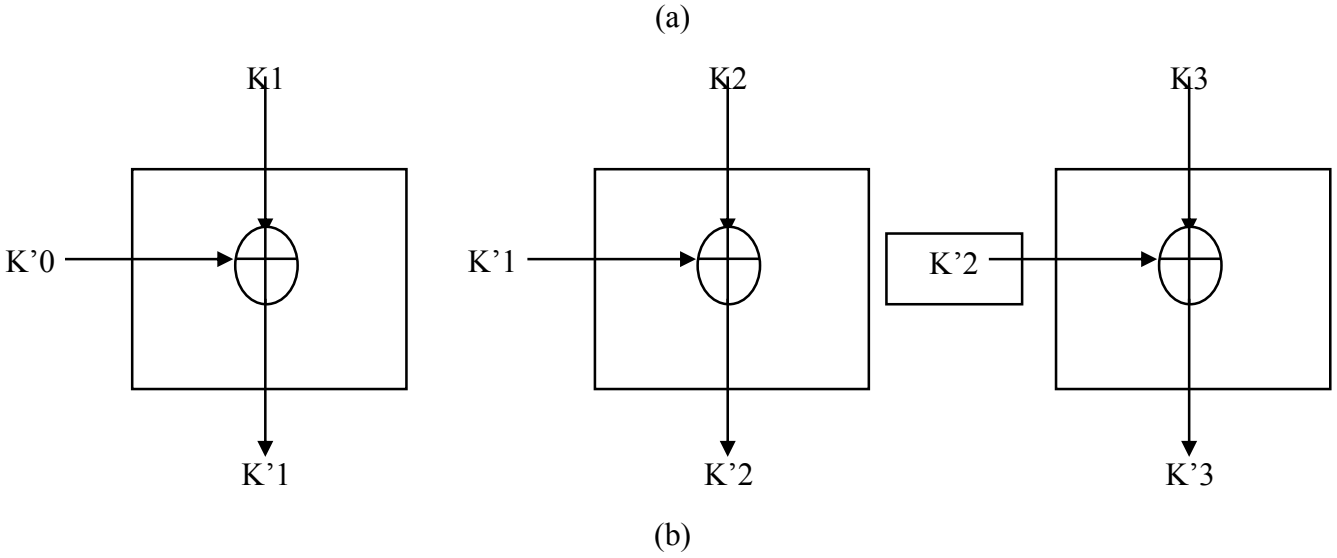


Fig. 2 Detailed hardware structure of round key generation phase (a) Dot block structure (b) Dot less block structure

Substitution bytes using CFA: Non-LUT based approaches for the sub bytes and its inverse is implemented using combinational circuits. This circuit calculates the values of the transformed byte on the fly without having the values pre-stored using tables. The implementation of a circuit to find the multiplicative inverse in the $GF(2^8)$ is very difficult and costly. By using CFA, an element in the higher order field $GF(2^8)$ is mapped to an element in the lower order field $GF(2^4)$ and can be further mapped to an elementary field. The multiplicative inverse can be found in the lower order field using the following polynomial equations [30],

$$(ax+b)^{-1} = a\left(a^2\lambda + b(a+b)\right)^{-1}x + (b+a)\left(a^2\lambda + b(a+b)\right)^{-1} \tag{5}$$

Inverse substitution bytes using CFA: It is performed by different collective operations, such as inverse affine transformation, isomorphic mapping, multiplicative inverse and inverse isomorphic mapping inverse affine transformation in matrix form is shown in the equation (6) and remains are similar to that equation.

$$A^{-1} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6}$$

The generated key from the modified key expansion approach is used for both block and stream ciphers and the process of generating keys "On the fly" reduces area but consumes time. This

Copyright ⓒ 2020 SERSC

can be overcome by combining this concept with the parallel sub-pipeline concept it is involved in main AES structure, as it saves time by doing the operation in high speed.

AES is an iterative round based symmetric key block cipher that supports key size of 128, 192, and 256 bits and block size of 128 bits. The use of larger key sizes increases the cryptographic strength but requires that greater number of iterative rounds be performed. Here we use 128 bit key size for AES implementation and it is enough to satisfy end users. Implementation of AES on hardware can be mainly divided into two modules, such as cipher module and key expansion module. Cipher module is responsible for performing encryption/decryption on the data while key expansion module is responsible for preparing the key that is required for each round of cipher. In case of 128 bit key, cipher module performs 10 rounds of substitution and permutation to transform the input data to ciphered data. For the first 9 rounds of encryption, cipher module makes use of sub-byte, shift-row, mix-column, add-round-key operations and for final round mix-column operation is skipped to complete the encryption process. Fig. 3 shows standard structure of AES algorithm. It can be seen from the figure that different functions of cipher module combined with key expansion module perform the encryption on input data through an iterative process. Input data in AES is often represented as 4×4 bytes array and output is cipher text such as X1, X2, X3 and X4.

The **sub-bytes** function performs a non-linear transformation independently on each byte of the input state. This transformation is performed by substituting each byte of the state with a value from substitution box (also termed as S-box). There are 16 parallel S-boxes each with 8 inputs and 8 outputs. The S-box operation is the only nonlinear transformation of AES algorithm. The detailed structure of non LUT based sub-byte and inverse sub-bytes are present in section 4.1.

The **shift-rows** function performs byte wise circular shifts on last three rows of the state. In this function, first row is not rotated, but second, third, and fourth rows are rotated by one, two, three bytes respectively. This rotation provides diffusion property of the AES algorithm. For the **inverse shift-rows** step, the second, third and fourth rows are shifted to the right by one, two and three offsets.

Mix-columns function separately operates on each of the four columns of states treating each column as a four-term polynomial. The columns are considered as polynomials over $GF\left(2^8\right)$ and multiplied modulo x^4+1 with a fixed polynomial $b(x)$ given as follows,

$$b(x)=\{03\}x^3+\{01\}x^2+\{01\}x+\{02\} \tag{7}$$

Above function can also be re written as matrix multiplication or in matrix form as follows,

$$\begin{bmatrix} a'_{0,c} \\ a'_{1,c} \\ a'_{2,c} \\ a'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} a_{0,c} \\ a_{1,c} \\ a_{2,c} \\ a_{3,c} \end{bmatrix} \tag{8}$$

Copyright ⓒ 2020 SERSC

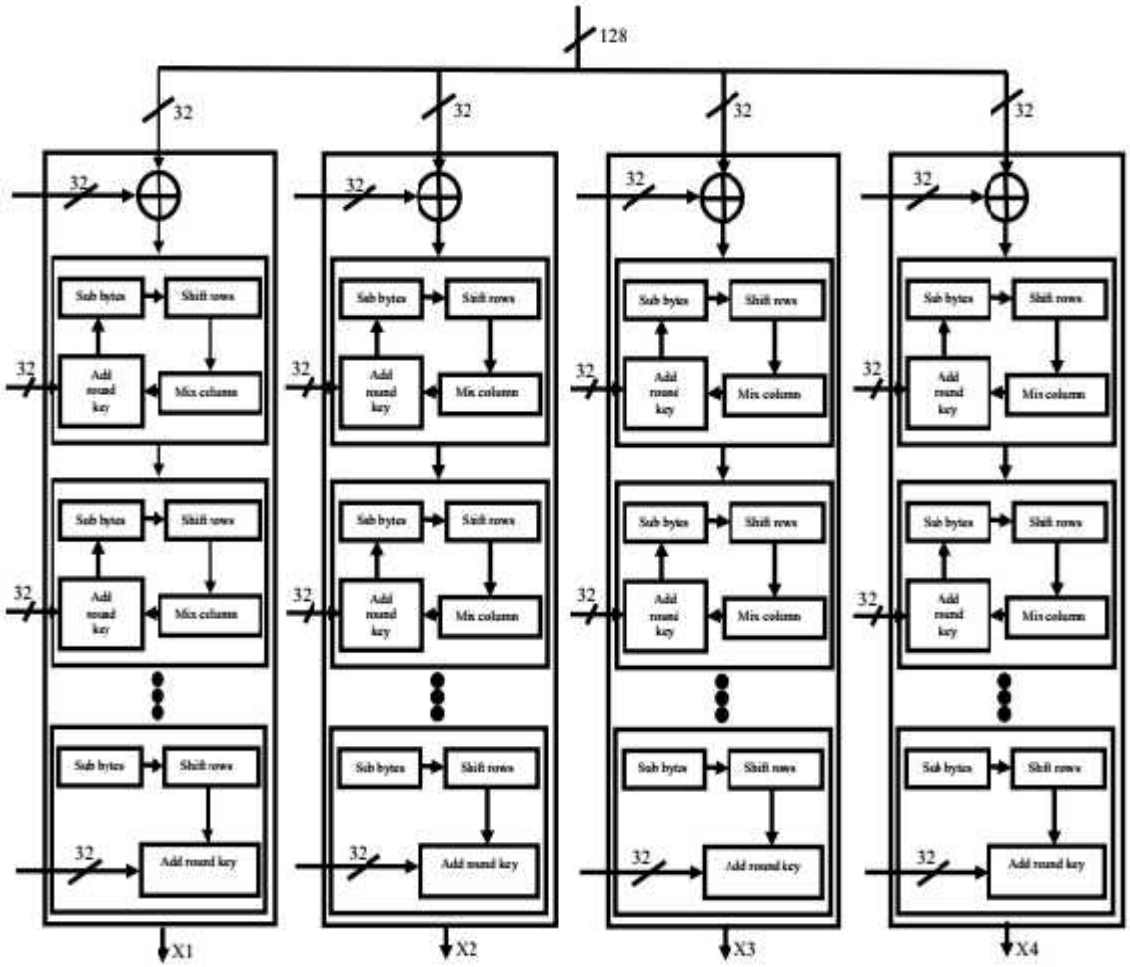


Fig. 3 parallel sub-pipeline structure of AES architecture

The **Inverse mix-column** step is expressed as follows,

$$\begin{bmatrix} a'_{0,c} \\ a'_{1,c} \\ a'_{2,c} \\ a'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} a_{0,c} \\ a_{1,c} \\ a_{2,c} \\ a_{3,c} \end{bmatrix} \quad (9)$$

Add-round-key is the final cipher function and is used to mix key information with the data that are being operated upon. Inputs of this function are 16 byte state and 16 byte key which is obtained from key expansion algorithm. Its output is a simple bit wise XOR operation between current round state and current round expanded key.

4.2 Side-channel resistant authenticated encryption with masking (SCREAM)

Generally, SCREAM ciphers consists 254 rounds and workable for 80-bit key size with variable block sizes: 32-, 48- and 64-bit. The algorithm start with initialization process, here, the plaintext is loaded into two registers R_1 and R_2 with the flexible length; and 80-bit key is also given to an

Copyright ⓒ 2020 SERSC

input. R_1 and R_2 register values are not similar for all the block size that is why we using flexible length registers. The register values are used to computes two nonlinear functions F_1 and F_2 on each round. The nonlinear functions are computes as follows:

$$F_1 = R_1[i_1] \oplus R_1[i_2] \oplus (R_1[i_3] \wedge R_1[i_4]) \oplus (R_1[i_5] \wedge IR) \oplus key_1 \quad (10)$$

$$F_2 = R_2[j_1] \oplus R_2[j_2] \oplus (R_2[j_3] \wedge R_2[j_4]) \oplus (R_2[j_5] \wedge R_2[j_6]) \oplus key_2 \quad (11)$$

IR is pre-computed irregular update rule and is the output of the most significant bit of the linear feedback shift register (LFSR). The parameters i and j represents the resister size which is not constant for all block sizes the details given in [38]. Key_1 and Key_2 are two sub key bits, for i-th iteration the keys denotes as $key_1 = key_{2x}$ and $key_2 = key_{2x+1}$. After 80 bits, the y-th bit of the key is generated as:

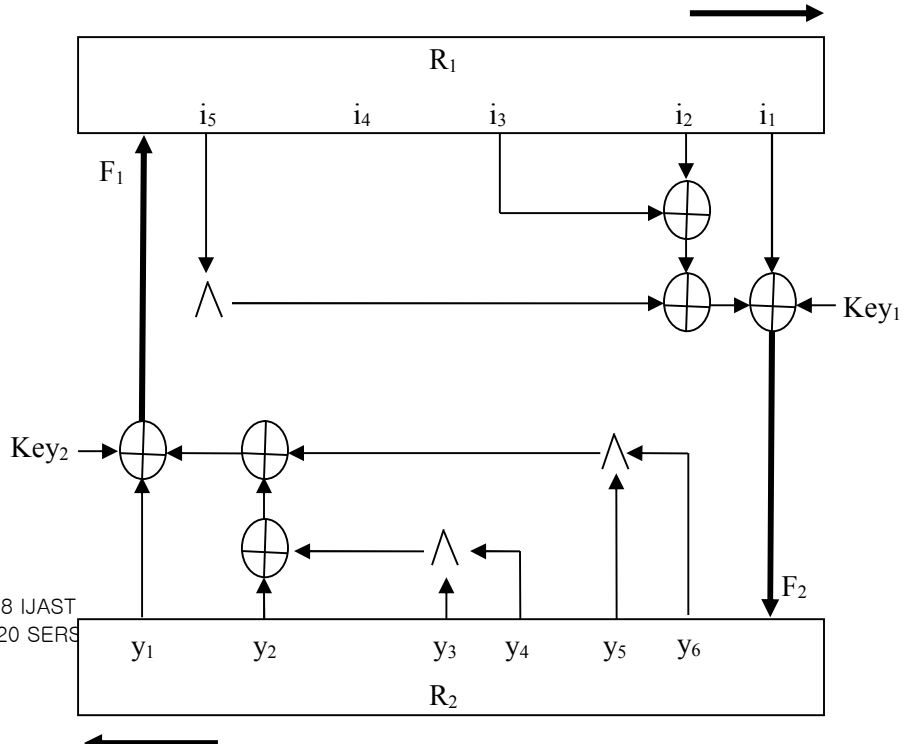
$$Key = \begin{cases} key_x; & for\ x = 0,1,2,\ldots 79 \\ key_y; & y = key_{y-80} \oplus key_{y-61} \oplus key_{y-50} \oplus key_{y-13} \end{cases} \quad (12)$$

In SCREAM ciphers, the computed non-linear functions F_1 and F_2 applied once, twice and thrice for SCREAM 32-bit, SCREAM 48-bit and SCREAM 64-bit respectively. The key consists of 5 words of 16 bits each as key $= w_4 \| w_3 \| w_2 \| w_1 \| w_0$. Then, sub key bits are computes as follows:

$$key_1 = \overline{RC_3} \wedge \overline{RC_2} \wedge mux_{16\times1}(w_0, RC_7RC_6RC_5RC_4) \oplus (RC_3 \vee RC_2) \wedge mux_{4\times1}(key_1, RC_1RC_0) \quad (13)$$

$$key_2 = \overline{RC_3} \wedge RC_2 \wedge mux_{16\times1}(w_4, RC_7RC_6RC_5) \oplus \left(RC_3 \vee \overline{RC_2}\right) \wedge mux_{4\times1}\left(key_2, \overline{RC_1RC_0}\right) \quad (14)$$

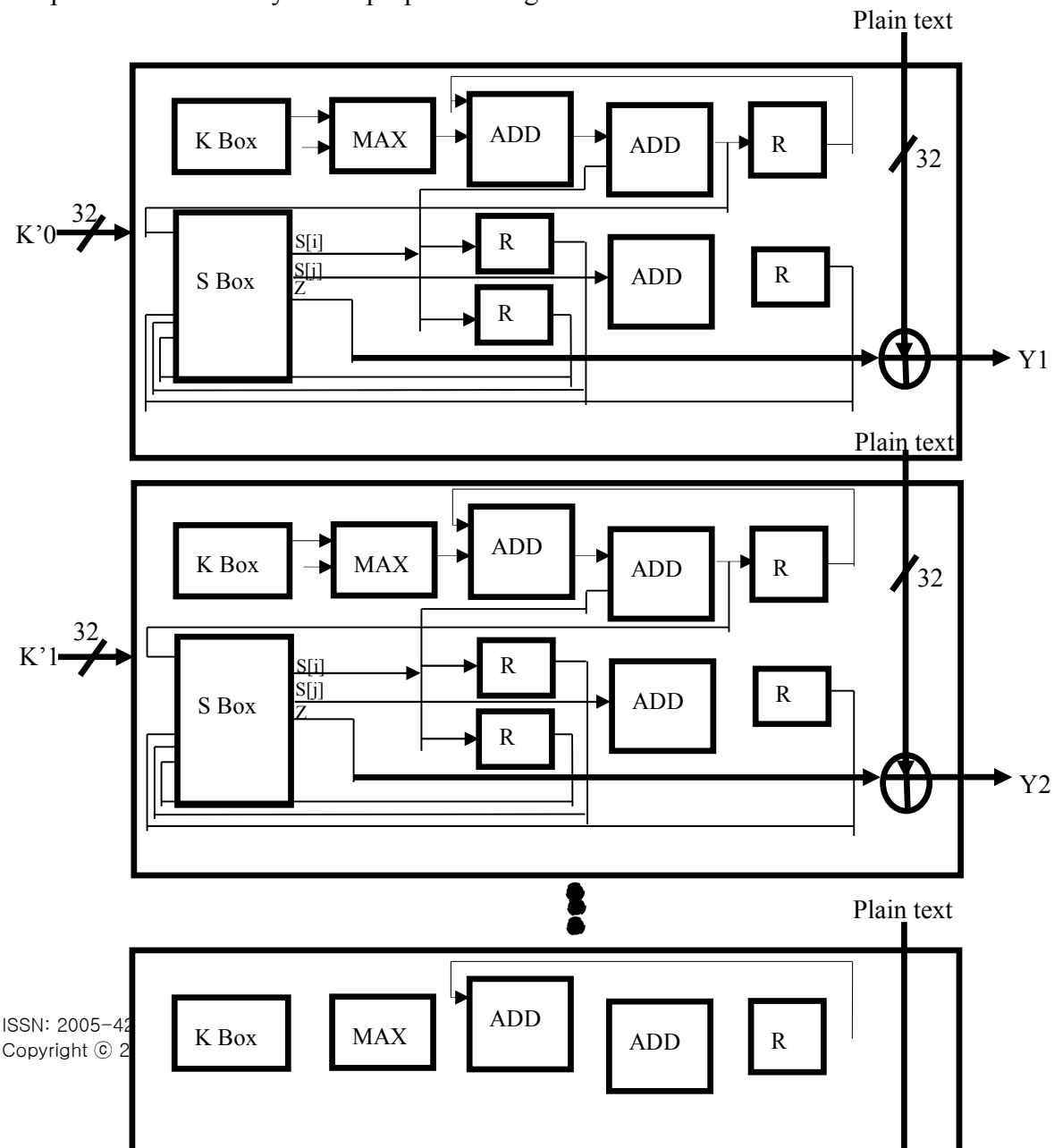
The round logic functional model is shown in Fig. 4 and it follows above mathematical equations. Our modified SCREAM ciphers use similar round logic with round logic controller in external manner and flexible register for reconfigurable avoidance.



Copyright ⓒ 2020 SERS

Fig. 4 Round logic module of SCREAM ciphers

The two bytes per clock pipeline [7] concept used to modify the pipeline structure of proposed design is shown in Fig. 5. The S_i computation and double swap operation starts at Stage 1 in this case, and takes the help of pipeline registers to maintain the read after-write ordering during the swap operations. This part of the operation is same as in the hardware pipelined approach for one-byte-per-clock design. The K and Z values are read from the registers after the completion of the S-box and rotation in AES and double-swap in RC4. That is, consecutive values of the output byte K and Z are read from the same state S by using some suitable combinational logic. Similarly, the increment of two consecutive i and j values are done simultaneously using the combinational logic of the original one-byte-per-clock design. This design obviously provides two output bytes per clock cycle, after an initial lag of one cycle. In HPSP_AES-RC4, we combine parallel and sub-pipeline ideas, and obtain the best throughput of two key expansions and key stream generations (AES and RC4) rounds per clock cycles. The total rounds are completed within 128 cycles in proposed design.



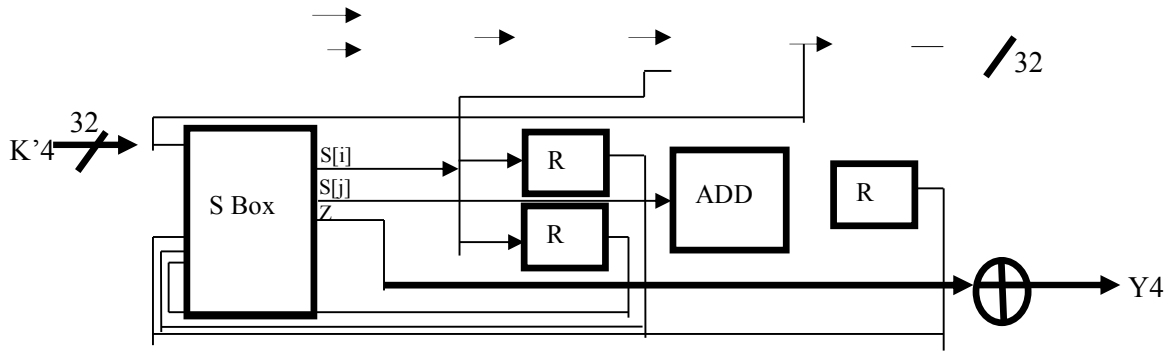


Fig. 5 Parallel sub-pipeline structure of RC4 stream cipher

The same clock cycles are used to complete key expansion/generation in single crypto system designs, for AES [16-26] and for SCAREM [27]. From this, the throughput of our hybrid cryptosystem same to the single cryptosystems [16-27], but the lifetime of generated key is increased.

5. Simulation Results

5.1 Security analysis

Due to the various possible attacks, AES was not fruitful in retrieving the key or the plain text. The authors in [26] prove that differential cryptanalysis and linear cryptanalysis performed on AES will not able to break and proves to be more secure. In algebraic attacks [33] which works such that the system is expressed as a multivariate polynomial equations which can be solved to find the key. The number of equations with thousands of unknown variables makes these less feasible for computing the key. In side channel attacks the variations in observable parameters are noted and cryptanalysis is done on these parameters. Only timing attacks are non invasive, all other side channel attacks like power analysis attack, fault injection attack, electromagnetic radiation are invasive. The probability of these attacks is normally less because of the requirement of precise measuring equipments and the requirement of encrypting device itself. Also there are many countermeasures available to overcome these side channel attacks, like increasing latency, masking of data, shuffling of data after every access etc. The attackers need to guess up to M times, where $M = 2^{64} \times 2^{33}(N-1), 1 \le N \le 10$ [26].

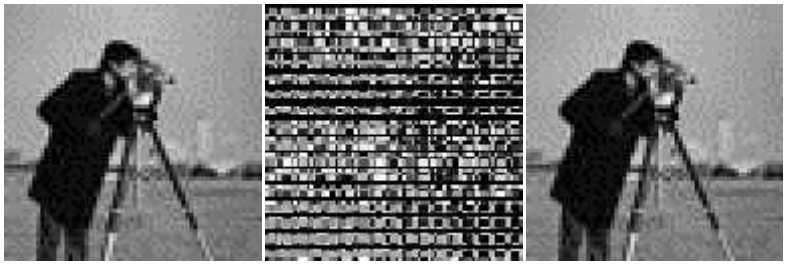
For our case, we implement hybrid algorithm which increase the key lifetime in terms of twice the previous case. The key lifetime of proposed cryptosystems as $2M$ and the attackers need to break the key up to 2M times. Due to this unbreakable key in the hybrid system, algebraic attacks are not possible to break the key with limited time. The proposed HWCP AES-SCREAM architecture is also tested for different set of input images. In this research, the gray scale images of different sizes are taken and are resized to 128×128 size. The gray scale images are converted to binary images in MATLAB. Now, the pixel values of the binary images are 1's and 0's which are then sent to the proposed hybrid encoder. In the encoder, the plain text/input bits are converted to the cipher/encrypted data. The encrypted data is again sent to the MATLAB using the FILE operation. The encrypted image's pixel values are then sent to the hybrid decoder. In the AES decoder, the encrypted data is converted to the decrypted data/plain text which is again linked to the MATLAB. The decrypted image obtained will be the binary image which is

Copyright ⓒ 2020 SERSC

again converted to gray scale image. Images of dog, lena, cameramen, peppers and ship are encrypted and decrypted using proposed architecture and are shown in fig. 6.

(a) (b) (c)

(a) (b) (c)



(a) (b) (c)

(a) (b) (c)

Copyright ⓒ 2020 SERSC

(a) (b) (c)

Fig. 6 Test images dog, lena, cameramen, peppers and ship with (a) Original image (b) Encrypted image (c) Decrypted image after reconstruction using proposed HPSP_AES-RC4 cryptosystem

5.2 Performance analysis

The performance of hardware structure of proposed design is discussed by area consumption, throughput and power consumption. The proposed HWCP architecture is prototyped in Xilinx Virtex XC6VLX75T device. The maximum clock frequency is reported by a Xilinx timing analyzer. The throughput demonstrates the impacts of pipelining and clock frequency. In addition, the number of slices used in the designs is shown as the measurement of implementation area, including used registers and LUTs. The maximum clock frequency and slices are obtained after synthesis and place and route, and the power consumption is reported by an Xpower Analyzer. The RTL schematic of the proposed HWCP architecture is shown in the Fig. 7. The hybrid parallel sub-pipeline architecture is proposed for maximizing the lifetime of key. In this architecture the 128 bit is divided into four bytes as shown in the Figure as input 1, input 2, input 3, and input 4. The encrypted and decrypted bits are shown as output from the block. Throughput is the speed at which the data is encrypted/decrypted. The throughput is very important in a communication process and this determines the performance of the algorithm. The throughput computation as follows:

$$Throughput = \frac{Block\ size \times Number\ of\ blocks\ per\ cycle}{Clock\ period} \tag{15}$$

Copyright ⓒ 2020 SERSC

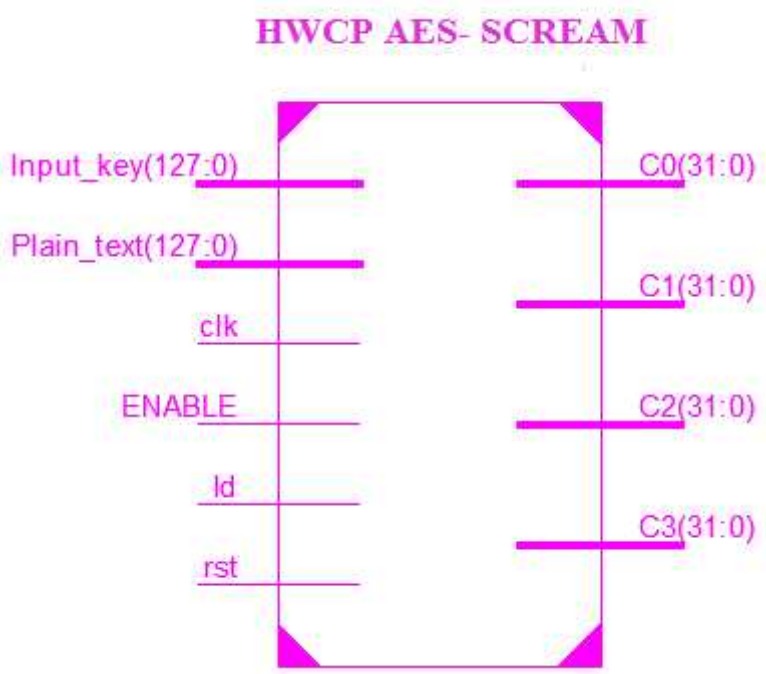


Fig. 7 RTL diagram of proposed HWCP AES- SCREAM

The performance comparison of the proposed HWCP AES- SCREAM architecture with the existing architecture reported in related works [16]-[25], [31] shown in the Table 1. The slice counts, maximum frequency and throughput are calculated for all the architectures in order to estimate and compare the speed of the algorithm. From the table, area utilization of our hybrid system is very little bit high compare to AES only cryptosystems. For hardware utilization, our hybrid system is efficient compare to previous works in and the throughput wise is more efficient compare to single cryptosystem. Some little deviation is there, but for hybrid system, it is efficient one. The power analysis of proposed architecture is compared with the existing AES architecture [31] and the comparison is tabulated in Table 2.

Copyright ⓒ 2020 SERSC

Table 1 Performance analysis comparison of proposed HWCP AES- SCREAM cryptosystem

Methods	Cryptosystem	FPGA family	Slices	Maximum Frequency	Throughput
[16]	AES-128 bit	Spartan-3 XC3S200	1385	202.02MHz	35068Mbps
		Virtex-4 XC4VLX200	1407	421.230MHz	54,008Mbps
[17]	AES-128 bit	Virtex-5	4491	333MHz	42.62Gbps
[18]	AES-128 bit	Virtex-7 XC7VX690T	8930	500MHZ	5548.65Mbps
[19]	REPROC	Stratix-IV EP4SE820	2136064 logic gates	400MHz	4.28Gbps
[20]	AES-128 bit	Virtex-5 XC5VLX85	28592	644.33MHz	82.4Gbps
		Virtex-6 XC6VLX240T	28520	803.98MHz	102.91Gbps
		Virtex-6 XC6VLX240T (CTR)	33328	508.104MHz	260.15Gbps
[21]	AES-128 bit	Virtex-5 XC5VLX85	2132	671.524MHz	86Gbps
[22]	ANF-CFA AES S-BOX	Cyclone-II EP2C5T144C6	95 logic elements	1436.3MHz	3.49Gbps

Copyright ⓒ 2020 SERSC

[23]	AESCCMP	Virtex-4 XC41X100-11	1921	149MHz	1.876Gbps
		Virtex-2 XC2V2000-6	1609	117.85MHz	1.484Gbps
		Spartan-3 XC3	1640	84.29MHz	1.0619Gbps
[24]	HMAC-Grøstl-256 and AES 128	Virtex-5	3102	235MHz	3848Mbps
		Virtex-6	2447	255MHz	4212Mbps
[25]	TSC SHA1 core	TSMC 90nm CMOS tech	-	563MHz	14.413Gbps
	TSC SHA 256 core	TSMC 90nm CMOS tech	-	503MHz	16.096Gbps
[31]	AES-128 bit	Vertex-7 XC7VX690T	4339	593.12MHz	75.92Gbps
		Vertex-6 XC6VLX240T	3900	573.39MHz	73.39Gbps
		Vertex-5 XC5VSX240T	4444	439.17MHz	56.21Gbps
		Vertex-5 XC5VLX110T	4445	403.39MHz	51.63Gbps
		Vertex-5 XC5VLX85	4447	420.35MHz	53.80Gbps
		Vertex-4 XC4VLX160	38511	454.55MHz	58.18Gbps
HWCP AES-SCREAM	Hybrid AES+SCREAM (128 bit)	Vertex-7 XC7VX690T	4924	591.23MHz	74.12Gbps
		Vertex-6 XC6VLX240T	3911	581.39MHz	74.65Gbps
		Vertex-5 XC5VSX240T	8966	431.79MHz	58.23Gbps
		Vertex-5 XC5VLX110T	8959	402.82MHz	50.97Gbps

Table 2 Power analysis comparison of proposed HWCP AES- SCREAM cryptosystem

FPGA family	Existing work power (W)	Proposed HPSP_AES-RC4 (W)
Vertex-7 XC7VX690T	3.59	0.289
Vertex-6 XC6VLX240T	8.08	0.564
Vertex-5 XC5VSX240T	7.56	0.612
Vertex-4 XC4VLX160	9.26	0.312

6. Conclusion

Copyright ⓒ 2020 SERSC

We have proposed a hardware efficient hybrid wireless crypto processor (HWCP), which combines two block ciphers such as enhanced advanced encryption standard (AES) and side-channel resistant authenticated encryption with masking (SCREAM). The hardware cost of hybrid processors are very high, here we use composite field arithmetic (CFA), on the fly key expansion, and order change to reduce the hardware parts in the encryption algorithms. The proposed HWCP design maximizes the security via increasing the complexity of cracking keys. Moreover, the proposed HWCP design is implemented with parallel sub-pipeline manner that increases the throughput. The results show that the performance of proposed HWCP design is very efficient than existing state-of-art architectures in terms of hardware utilization, power consumption and maximum operating frequency.

References

[1]Schuster, D.B., 2014. International telecommunication union-Challenges for the plenipotentiary 2014-The time for change. IEEE Communications Magazine, 52(2), pp.57-61.

[2]Shan, W., Zhang, S. and He, Y., 2017. Machine learning based side-channel-attack countermeasure with hamming-distance redistribution and its application on advanced encryption standard. Electronics Letters, 53(14), pp.926-928.
[3]Yang, K., Yu, Q., Leng, S., Fan, B. and Wu, F., 2016. Data and energy integrated communication networks for wireless big data. IEEE access, 4, pp.713-723.

[4] Liu, T.Y., Lin, S.C. and Hong, Y.W.P., 2016. On the role of artificial noise in training and data transmission for secret communications. IEEE Transactions on Information Forensics and Security, 12(3), pp.516-531.

[5]Jang, T., Kim, G., Kempke, B., Henry, M.B., Chiotellis, N., Pfeiffer, C., Kim, D., Kim, Y., Foo, Z., Kim, H. and Grbic, A., 2017. Circuit and system designs of ultra-low power sensor nodes with illustration in a miniaturized GNSS logger for position tracking: Part II—Data communication, energy harvesting, power management, and digital circuits. IEEE Transactions on Circuits and Systems I: Regular Papers, 64(9), pp.2250-2262.

[6]Nalawade, S.B. and Gawali, D.H., 2017, October. Design and implementation of blowfish algorithm using reconfigurable platform. In 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE) (pp. 479-484). IEEE.

[7]Clulow, J., 2003, September. On the security of PKCS# 11. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 411-425). Springer, Berlin, Heidelberg.

[8]Shan, W., Chen, X., Li, B., Cao, P., Li, J., Gao, G. and Shi, L., 2013. Evaluation of correlation power analysis resistance and its application on asymmetric mask protected data encryption standard hardware. IEEE Transactions on Instrumentation and Measurement, 62(10), pp.2716-2724.

[9]Huang, X. and Wang, W., 2015. A novel and efficient design for an RSA cryptosystem with a very large key size. IEEE Transactions on Circuits and Systems II: Express Briefs, 62(10), pp.972-976.

Copyright ⓒ 2020 SERSC

[10]Ara, A., Al-Rodhaan, M., Tian, Y. and Al-Dhelaan, A., 2017. A secure privacy-preserving data aggregation scheme based on bilinear ElGamal cryptosystem for remote health monitoring systems. IEEE Access, 5, pp.12601-12617.

[11]Pan, M., Sun, J. and Fang, Y., 2011. Purging the back-room dealing: Secure spectrum auction leveraging paillier cryptosystem. IEEE Journal on Selected Areas in Communications, 29(4), pp.866-876.

[12]Joshi, N., Sundararajan, J., Wu, K., Yang, B. and Karri, R., 2006. Tamper proofing by design using generalized involution-based concurrent error detection for involutional Substitution Permutation and Feistel Networks. IEEE Transactions on Computers, 55(10), pp.1230-1239.

[13]Guoqiang, B., Zhun, H., Hang, Y., Hongyi, C., Ming, L., Gang, C., Tao, Z. and Zhihua, C., 2004, October. A high performance VLSI chip of the elliptic curve cryptosystems. In Proceedings. 7th International Conference on Solid-State and Integrated Circuits Technology, 2004. (Vol. 3, pp. 2059-2062). IEEE.

[14]Gutub, A.A., 2007. High speed hardware architecture to compute galois fields GF (p) montgomery inversion with scalability features. IET Computers & Digital Techniques, 1(4), pp.389-396.

[15]Kwok, S.H. and Lam, E.Y., 2008. Effective uses of fpgas for brute-force attack on rc4 ciphers. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 16(8), pp.1096-1100.

[16] Yicheng, C., Xuecheng, Z., Zhenglin, L., Yu, H. and Zhaoxia, Z., 2008. Energy-efficient and security-optimized AES hardware design for ubiquitous computing. Journal of Systems Engineering and Electronics, 19(4), pp.652-658.

[17]Rahaman, H., Mathew, J. and Pradhan, D.K., 2010. Secure testable s-box architecture for cryptographic hardware implementation. The Computer Journal, 53(5), pp.581-591.

[18]Cilardo, A. and Mazzocca, N., 2013. Exploiting vulnerabilities in cryptographic hash functions based on reconfigurable hardware. IEEE Transactions on Information Forensics and Security, 8(5), pp.810-820.

[19]Monteiro, C., Takahashi, Y. and Sekine, T., 2015. Low-power secure S-box circuit using charge-sharing symmetric adiabatic logic for advanced encryption standard hardware design. IET Circuits, Devices & Systems, 9(5), pp.362-369.

[20]Wu, N., Zhang, X.Q. and Yahya, M.R., 2016. Highly optimised reconfigurable hardware architecture of 64 bit block ciphers MISTY1 and KASUMI. Electronics Letters, 53(1), pp.10-12.

[21]Liu, Q., Xu, Z. and Yuan, Y., 2015. High throughput and secure advanced encryption standard on field programmable gate array with fine pipelining and enhanced key expansion. IET Computers & Digital Techniques, 9(3), pp.175-184.

Copyright ⓒ 2020 SERSC

[22]Kundi, D.S., Aziz, A. and Ikram, N., 2016. A high performance ST-Box based unified AES encryption/decryption architecture on FPGA. Microprocessors and Microsystems, 41, pp.37-46.

[23]Constantin, J., Houlmann, R., Preyss, N., Walenta, N., Zbinden, H., Junod, P. and Burg, A., 2017. An FPGA-based 4 Mbps secret key distillation engine for quantum key distribution systems. Journal of Signal Processing Systems, 86(1), pp.1-15.

[24]Subramanian, S., Mozaffari-Kermani, M., Azarderakhsh, R. and Nojoumian, M., 2017. Reliable hardware architectures for cryptographic block ciphers LED and HIGHT. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 36(10), pp.1750-1758.

[25]Akleylek, S., Alkım, E. and Tok, Z.Y., 2016. Sparse polynomial multiplication for lattice-based cryptography with small complexity. The Journal of Supercomputing, 72(2), pp.438-450.

[26]Wang, B. and Liu, L., 2016. Dynamically reconfigurable architecture for symmetric ciphers. Science China Information Sciences, 59(4), p.042403.

[27]Granado-Criado, J.M. and Vega-Rodríguez, M.A., 2017. Hardware coprocessors for high-performance symmetric cryptography. The Journal of Supercomputing, 73(6), pp.2456-2482.

[28]Aziz, A., 2016. A low-power SHA-3 designs using embedded digital signal processing slice on FPGA. Computers & Electrical Engineering, 55, pp.138-152.

[29]Farooq, U. and Aslam, M.F., 2017. Comparative analysis of different AES implementation techniques for efficient resource usage and better performance of an FPGA. Journal of King Saud University-Computer and Information Sciences, 29(3), pp.295-302.

[30]Vollala, S., Varadhan, V.V., Geetha, K. and Ramasubramanian, N., 2017. Design of RSA processor for concurrent cryptographic transformations. Microelectronics Journal, 63, pp.112-122.

Copyright ⓒ 2020 SERSC