International Journal of Future Generation Communication and Networking
Vol. 13, No. 2s, (2020), pp. 38–47

Secure Encryption of data using Symmetric and Asymmetric Cryptographic Algorithm

Sreyash Shrivastava[1], Smita Palnitkar[2], Sumit Pathak[3], Mahesh Shinde[4]
[1,2,3,4] *Dept. of E & TC Engg., Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, Pune*
[1]*sreyash810@gmail.com*
[2]*smitpalnitkar@gmail.com*
[3]*sumitpathak6@gmail.com*
[4]*maheshshinde787@gmail.com*

Abstract

Encryption is the process of transforming information into an unidentifiable code so that only the intended recipient can access it and those who are not authorized cannot. Encryption does not itself prevent interference, but denies the intelligible content to a would-be interceptor. It can still be used to protect a user's identity and privacy. As more and more information is stored on computers or communicated via computers, the need of the hour is to ensure that this information is secure and to prevent from snooping / tampering becomes more relevant. Even if data does end up getting stolen, it will be unreadable and nearly useless if it's encrypted. With the fast progression of digital data exchange in electronic way, Encryption is essential for ensured and trusted delivery of sensitive information sent over the internet. Contemporary human being intelligence, lead to cryptography has become more complex in order to make information more secure. Meanwhile many encryption algorithms are being developed in the world of Cyber security society. In this paper, a hardware implementation of three standard cryptography algorithms on a universal architecture.The cryptography processor algorithm targets smart card applications and implements both private and public-key algorithms.

Keywords*—Information Security, Encryption, DES, ECC, AES.*

I. Introduction

The rapid growth of portable electronic devices with limited power and area has opened a vast area of low-power and compact circuit design opportunities and challenges for VLSI circuit designers. Cellular phones, PDAs, and smart cards are examples of portable electronic products that are becoming an integral part of everyday life. The popularity of these devices necessitates special considerations for their security subsystems. Unlike computer network security systems that impose less stringent limitations on the area and power consumption but put more emphasis on high throughput (several Gigabit/s), portable applications demand security hardware with more restrictions on area and power and less on throughput (several hundred kilobit/s to a few Megabit/s). This difference in requirements dictates a different approach in the design and implementation of the security systems for these devices.

Private Key algorithms with high throughput are suitable for data communication, while public key algorithms with much lower throughput are suitable for private key exchange and authentication. Among all available algorithms, data encryption standard (DES), advanced encryption standard (AES), and elliptic curve cryptography (ECC), which are approved by standards organizations, are selected for this application. DES, for past compatibility, and AES, for high security and throughput, are the major candidates for private key algorithms, and ECC is the best candidate for the public key algorithm for its encryption efficiency. RSA, which is also a standard public key algorithm, is not considered in this design for three reasons. First, it is believed that 160-b ECC

ISSN: 2233-7857 IJFGCN

provides the same level of security as 1024-b RSA. Thus, ECC will be a better choice when implementation area is a key factor in the design. Second, RSA uses binary addition of large numbers and needs binary adders that are either slow for carry propagation or large for look-ahead carry generation. Third, a larger number of bits in RSA means wider buses, which adds to the area and power consumption of the design, both of which are scarce resources in smart cards.

A cryptography system can be implemented in either soft- ware or hardware. Software implementations allow multiple algorithms to be supported on the same hardware platform, but they are usually slow and cannot meet the required specifications. Moreover, they are considered to be more vulnerable to side-channel attacks compared to other implementations. Side-channel attacks use physical measurements on the device, for example, the power consumption of the processor, to detect the encryption/decryption key .On the other hand, hard- ware implementations which support high throughput do not allow for flexibility and, hence, are not suitable for smart cards.

RISC processors that implement the cryptography algorithms in software provide a very attractive throughput, area, and power consumption solution. However, since users and manufacturers are still concerned about the vulnerability of software implementations of the cryptography algorithms to side-channel attacks, it is important to design hardware circuits that meet the required specifications and are more immune to these attacks. Considering all of the limitations discussed so far, it is quite challenging to de- sign and implement an algorithm-agile and area-and-power-efficient crypto-processor for smart cards with acceptable performance and security. The published works in this area address only a subset of these issues. References and consider ECC only, implements DES, and implements AES only. The FPGA implementation in is a processor that does not support AES and ECC.

II. LITERATURE SURVEY

S. S. Raghuram and C. Chakrabarti, “A programmable processor for cryptography,” in Proc. ISCAS, pp. V-685–V-688.[1]
In this paper, a programmable architecture that can handle a large number of algorithms including DES, RSA, Blowfish, SAFER, et cetera has been developed. The architecture consists of addition, subtraction, modular multiplication, and exponentiation and XOR units and thus can support a majority of the cryptographic algorithms. A high data rate is achieved by applying loop unrolling to the Montgomery algorithm that is used for modular multiplication and exponentiation. The differences in the number of bits, key length, and sequence of operations are handled by the micro programmed control unit. A VHDL model has been developed and synthesized using Auto Logic I1 from Mentor Graphics. The results show a frequency of operation of 77 Megahertz and an area of 23,000 “Optimization COST” units.

M. Aydos, T. Yanik, and C. K. Koc, “High-speed implementation of an ECC-based wireless authentication protocol on an ARM microprocessor,” Proc. IEE Commun., vol. 148, no. 5, pp. 273–279, Oct. 2001[2]
This paper present the results of implementation of elliptic curve cryptography (ECC) over the field GF(p) on an 8O-MHZ, 32- bit ARM microprocessor: The project produced a practical software library which supports variable length implementation of the elliptic curve digital signature algorithm (ECDSA). Paper implemented the ECDSA and a recently proposed ECC-based wireless authentication protocol using the library. The timing results show that the 160-bit ECDSA signature generation and verification operations take around 46 ms and 94 ms, respectively. With these

ISSN: 2233-7857 IJFGCN

timings, the execution of the ECC-based wireless authentication protocol takes around 140 ms on the ARM7TDMI processor, which is a widely used, low-power core processor for wireless applications.

J.Goodman and A.P. Chandrakasan, "An energy-efficient reconfigurable public-key cryptography processor," IEEEJ. Solid- State Circuits, vol. 36, no.11, pp. 1808–1820, Nov. 2001.[3]
This paper presents a domain-specific configurability which is utilized to provide the required flexibility, without incurring the high overhead costs associated with generic reprogrammable logic. The resulting implementation is capable of performing an entire suite of cryptographic primitives over the integers modulo N, binary Galois Fields and non-super singular elliptic curves over GF (2^8), with fully programmable moduli, field polynomials and curve parameters ranging in size from 8 to 1024 bits. The resulting processor consumes a maximum of 75 mW when operating at a clock rate of 50 MHz and a 2-V supply voltage. In ultralow-power mode (3 MHz at 0.7 V) the processor consumes at most 525uW. Measured performance and energy efficiency indicate a comparable level of performance to previously reported dedicated hardware implementations, while providing all of the flexibility of a software-based implementation. In addition, the processor is two to three orders of magnitude more energy efficient than optimized software and reprogrammable logic.

E. Trichina, M. Bucci, D. De Seta, and R. Luzzi, "Supplemental cryptographic hardware for smart cards," IEEE Micro, pp. 26– 35, Nov.–Dec. 2001[4]
The author describe true random number generator and an AES computational unit; two different special purpose hardware block that can be integrated on the silicon substrate along with the usual functional unit of a smart card microcontroller. The paper synthesized the Randaes logic using the Ambit synthesis tool from Cadence with a 0.18-micron standard-cell library. The synthesis results furnished an area of approximately 0.025 mm2 for RNGA, 0.016 mm2 for RNGB, and 0.002 mm2 for RNGC. The postprocessor's total area is 0.04 mm2. Their required throughput for RNGs after post processing is from 2.5 to 10 Mbps. The synthesis result for AES furnished an area of approximately 0.1 mm2. With the clock period fixed at 3 ns, the coprocessor achieves approximately 24-Mbps throughput. A total ciphering time (that is, time required for generating round key and enciphering one block) is 18 microseconds.

P.H.W.Leong and I.K.H. Leung, "A microcoded elliptic curve processor using FPGA technology," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 10, no. 5, pp. 550–559, Oct. 2002[5].
The implementation of a micro coded elliptic curve processor using field-programmable gate array technology is described. This processor implements optimal normal basis field operations in $F(2^m)$. The design is synthesized by a parameterized Module generator, which can accommodate arbitrary n and also produce field multipliers with different speed/area trade-offs. The control part of the processor is micro coded, enabling curve operations to be incorporated into the processor and hence reducing the chip's I/O requirements. The micro coded approach also facilitates rapid development and algorithmic optimization: for example, projective and affine coordinates were supported using different microcode. The design was successfully tested on a Xilinx Virtex XCV1000-6 device and could perform an elliptic curve multiplication over the field F2m using affine and projective coordinates for n = 113, 155 and 173.

I.Verbauwhede, P. Schaumont, and H. Kuo, "Design and performance testing of a 2.29-GB/s rijndael processor," IEEE J. Solid- State Circuits, vol. 38, no. 3, pp. 569–572, Mar. 2003.[6]
This paper describes the design and performance testing of an Advanced Encryption Standard (AES) compliant encryption chip that delivers 2.29 GB/s of encryption throughput at 56 mW of power consumption in a 0.18-um CMOS standard cell technology. This integrated circuit

ISSN: 2233-7857 IJFGCN

implements the Rijndael encryption algorithm, at any combination of block lengths (128, 192, or 25 bits) and key lengths (128, 192, or 256 bits). We present the chip architecture and discuss the design optimizations. We also present measurement results that were obtained from a set of 14 test samples of this chip. This project can reduce the gate count as well as the critical path to half the current amount.

N. S. Kim, T. Mudge, and R.Brown, "A 2.3 Gb/s fully integrated and synthesizable aes rijndael core," in Proc. IEEE Custom Integrated Circuits Conf., 2003, pp. 193–196.[7]
In this paper, they presented a fully integrated and synthesizable cipher core supporting the Advanced Encryption Standard - Rijndael. They designed and fabricated the fully integrated core - key scheduler, encipher, and decipher using TSMC 0.18pm technology. The core operating frequency is 465MHz and throughput is 2.3Gbis. The chip can process all the standard key sizes and plain text lengths - 128, 196, and 256 bits. The design was fabricated with TSMC 0.18pm process and the tested result shows that the maximum throughput is 2.33Gb/s with a power consumption of 0.314W. They also proposed the round key cache to improve key scheduling efficiency. It takes advantage of the temporal locality of the cipher key in cryptographic.

III. IMPLEMENTATION DETAILS

Our procedure for designing this universal algorithm starts by listing all three algorithms with a minimum set of required blocks for each algorithm. Then, a cost function, which is the area in this case, but could be any other parameter such as power or speed or any combination of several parameter, is defined and calculated for the selected set. If the cost is larger than the available budget, the complex functional blocks are expressed in terms of simpler blocks by revising the algorithm primitives or by selecting a new mathematical interpretation of the operation. This should be followed by the selection of the minimum set of functional blocks and calculation of the new cost. The procedure should be repeated until the cost criterion is met or the simplest form of the functional blocks is achieved.

TDES
The encryption/decryption process is equivalent to performing three standard DES encryption/decryption sequences. The Standard Triple-DES core iteratively performs the DES encryption/decryption process. Fast Triple-DES implementation contains three DES modules arranged in a pipelined architecture. Thus, the first output block is obtained in three times the time needed by a single DES core. However, each subsequent input block can be fed to the core in the time needed for a single DES encryption/decryption. If the input feed to the core is maintained constant in this respect, each output block, save the first, is obtained after time needed
to perform single DES processing. The parameters used in the main flow of DES depicted are 32 b wide and the implementation of the operations in each round is straightforward. Note that the key scheduler is also modified to have its output match the changes in the f()-box.

ISSN: 2233-7857 IJFGCN

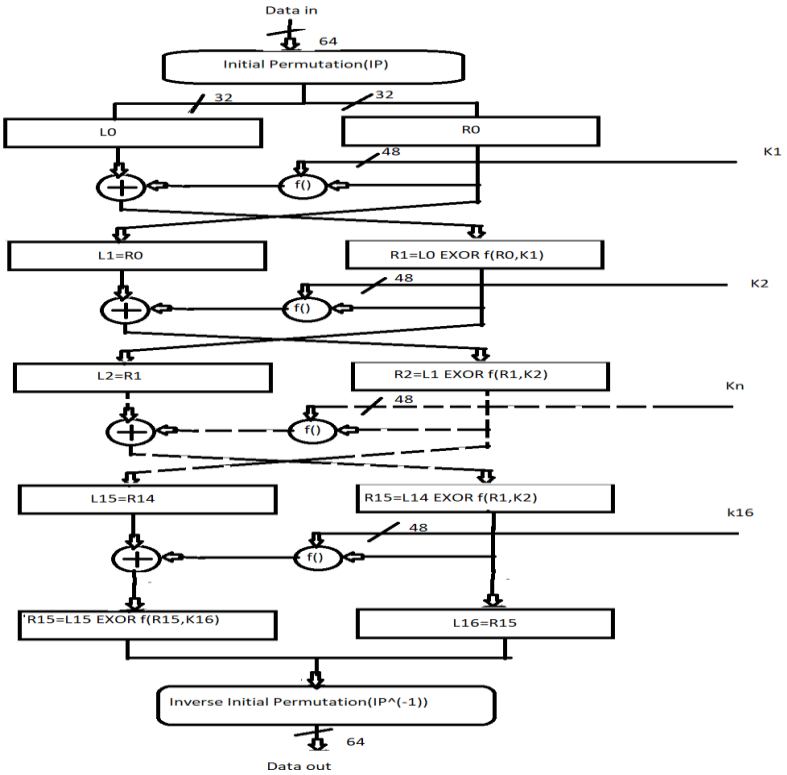


Fig. 1 TDES Encryption technique

AES

1) Key Expansions round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more. 2) Initial Round:

2) AddRoundKey each byte of the state is combined with a block of the round key using bitwise xor.

3) Rounds

- Sub Bytes
- Shift Rows
- Mix Columns
- AddRoundKey

4) Final Round (no Mix Columns)

a) Sub Bytes

b) Shift Rows

c) AddRoundKey

AES is an iterated block cipher with a fixed block size of 128 and a variable key length.

The different transformations operate on the intermediate results, called state. The state is a rectangular array of bytes and since the block size is 128 bits, which is 16 bytes, the rectangular array is of dimensions 4x4. (In the Rijndael version with variable block size, the row size is fixed to four and the number of columns varies. The number of columns is Advanced Encryption

Standard (AES) the block size divided by 32 and denoted Nb). The cipher key is similarly pictured as a rectangular array with four rows. The number of columns of the cipher key, denoted Nk, is equal to the key length divided by 32.

ISSN: 2233-7857 IJFGCN

PRE ROUND OPERATION

In this operation, a given data input (128 bits) is bitwise XORed with User defined key (128 bits) to generate a cipher text of 128bits.

Example:

Input = 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34

Cipher Key = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c Output =19 3d e3 be a0 f4 e2 2b 9a c6 8d 2a e9 f8 48 08

THE SUB-BYTES STEP:

Sub-Bytes operation is a non-linear byte substitution, operating on each byte of the state independently. The substitution table (S-Box) is invertible and is constructed by the composition of two transformations:

1. Take the multiplicative inverse in Rijndael's finite field
2. Apply an affine transformation as described as b′i = bi+b(i+4) mod8+b (i+5) mod8+b(i+6) mod8+b(i+7) mod8+ci for 0 <i< 8

, where bi is the ith bit of the byte, and ci is the ith bit of a byte c with the value {63} or {01100011}. Here and elsewhere, a prime on a variable indicates that the variable is to be updated with the value on the right.

SHIFT ROW OPERATION

In this operation, each row of the state is cyclically shifted to the left, depending on the row index. The 1st row is shifted 0 positions to the left. The 2nd row is shifted 1 position to the left. The 3rd row is shifted 2 positions to the left. The 4th row is shifted 3 positions to the left.

MIX COLUMN OPERATION

The Mix Columns transformation operates on the State column-by-column, treating each column as a four-term polynomial as described in Sec.2.2.5. The columns are considered as polynomials over GF (28) and multiplied modulo x4 + 1 with a fixed polynomial a(x), given by

a(x) = {03}x3 +

{01}x2 + {01}x +

{02} KEY

GENERATOR

OPERATION

The key generator circuit functions to generate unique key for every round operation in AES algorithm. Key expander (or
generator) operation basically follows five steps to generate a unique key for each round. User defined is fed as an input to Key expander circuit to find the key generated output.

ADD ROUND KEY OPERATION

ISSN: 2233-7857 IJFGCN

The primary function of Add Round Key Operation is to associate key-expander output generated by key generator Circuit to the AES algorithm. In this operation, a Round Key is applied to the state by a simple bitwise XOR. The Round Key is derived from the Cipher Key by the means of the key schedule. The Round Key length is equal to the block key length (=16 bytes). Add round key Output is given by XORing of Key-exp output and Mix-column output. The above output is*the encrypted* output of round 1. The Add round key output is again feedback to the Sub Byte transformation through feedback loop for 2nd round of operation end the same process is repeated until it completes 10 rounds of operation.

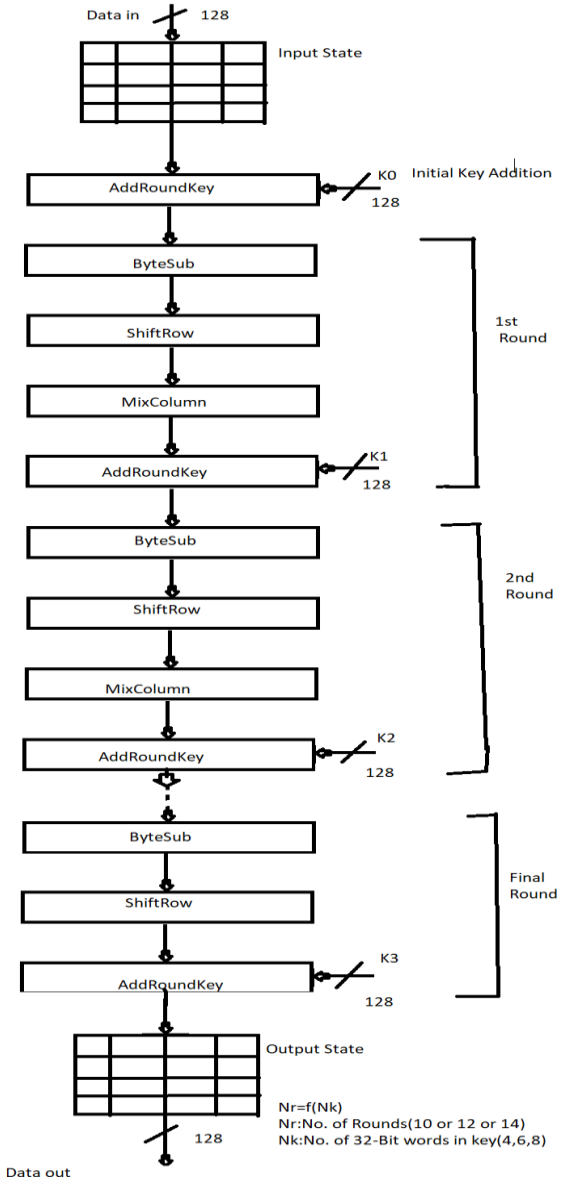


Fig .2 AES Encryption technique

ISSN: 2233-7857 IJFGCN

ECC

The point multiplication $Q = K. P$ in ECC is the basic operation of this algorithm, and, instead of k times the addition of the point P to itself, it is more efficient to use the binary expansion of k and use a set of doubling and addition to find Q ,as presented in[11]. Field addition and doubling is cheap (XOR and Circular shift left) and is done in the point-multiplication main routine. It should be emphasized that, since all of the computations are performed in a single basis (ONB), there is no need for the basis conversion, and hence no basis conversion overhead is associated with this design.

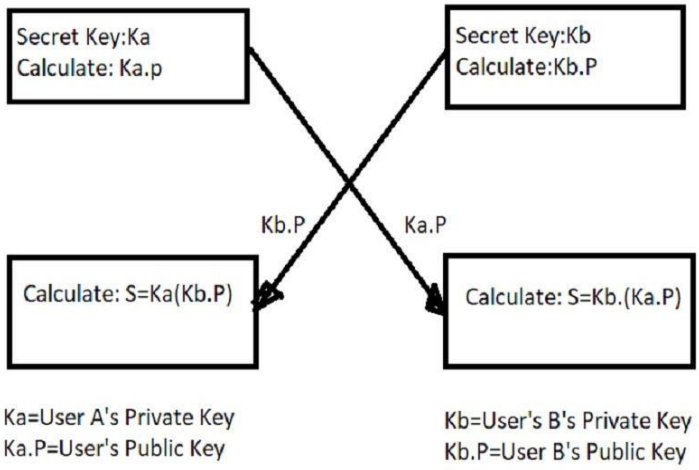


Fig. 3 ECC Key Exchange technique

Flow-chart

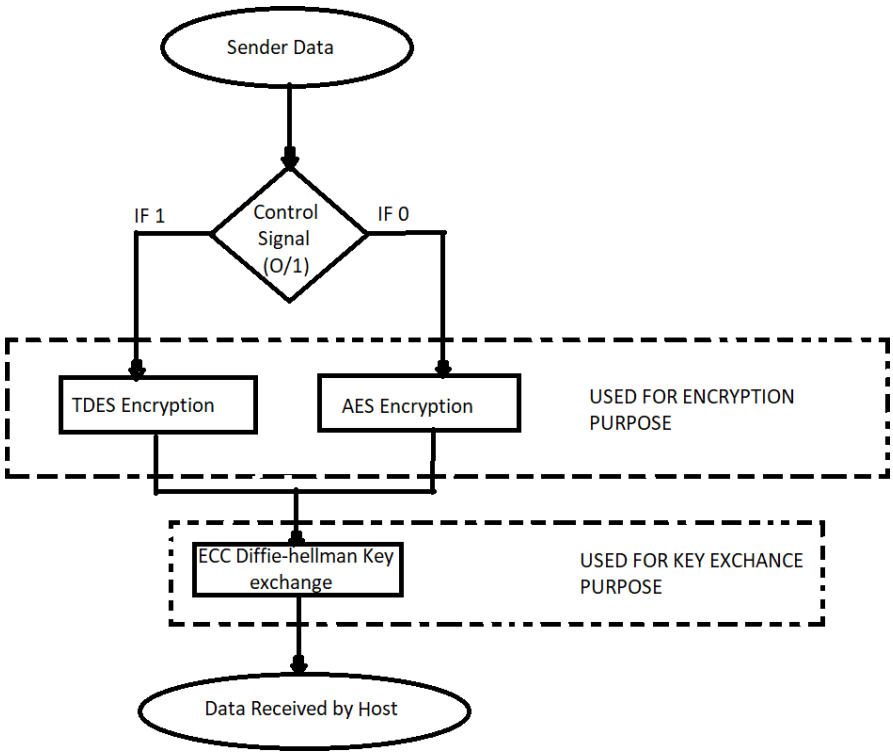


Fig. 4 Flow chart of Cryptography algorithm

ISSN: 2233-7857 IJFGCN

IV. DESIGN PERFORMANCE

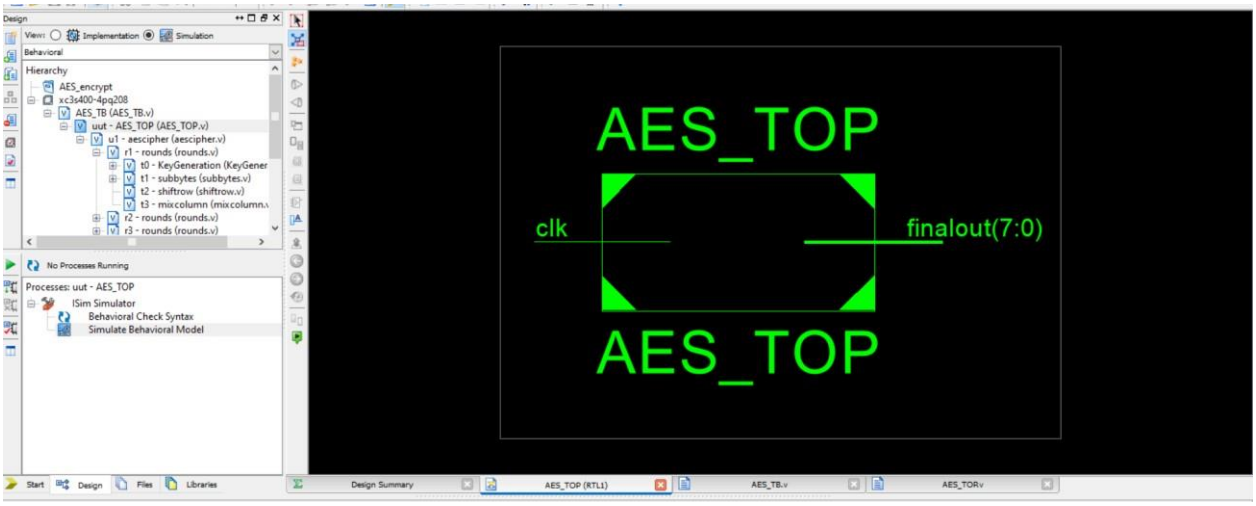
We used Verilog-XL to simulate both the RTL and gate-level netlists of the designed cryptography algorithm. the throughput of encryption and decryption of DES and encryption of AES algorithms are much higher than the maximum data transfer rate needed, hence, for more power-restricted applications, the clock frequency can be lowered for these cases to reduce the power consumption.

Also note that, for ECC, the performance is measured in point multiplication per second (PM/s) rather than bits per second (b/s), because ECC is mainly used for secret key exchange and authentication purposes which both use point multiplication as the basic operation. The ECC performance plot shows that highly secure applications(23*b*,92*b*,54*b* ECC) need to spend around 1.5s for secret key exchange or a point multiplication, while medium (131,155b)and low(83b) security applications can perform the same operations in less than a second. Moreover, it is possible to trade performance with area and power in this implementation. For example, higher performance can be obtained by running the algorithm at higher frequencies—up to 25 MHz for the current design—(increasing power consumption) and/or using pipelining (increasing area), for more performance-demanding applications.

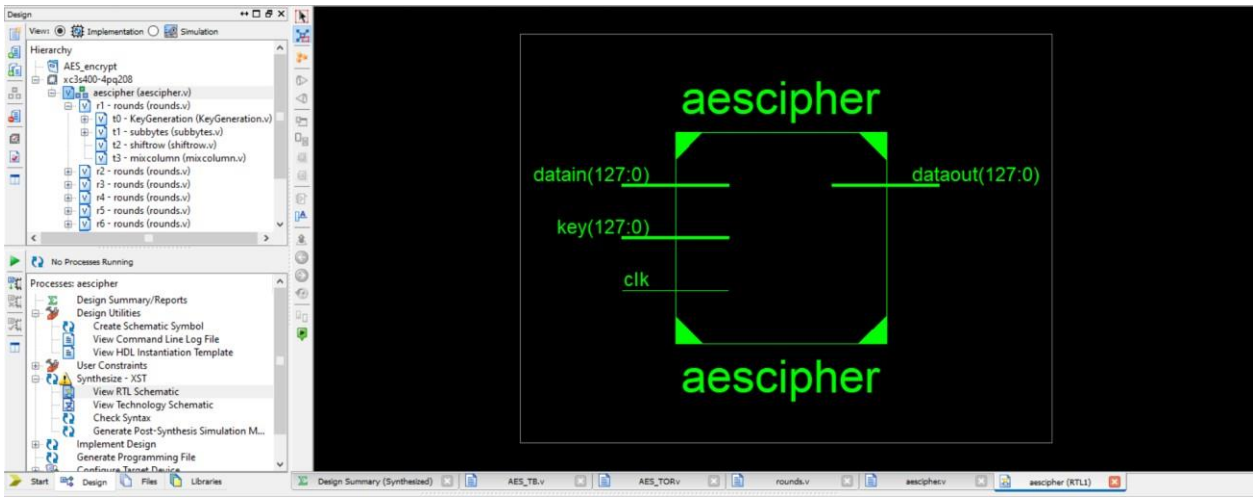
V. SIMULATION RESULTS

AES

1) S-BOX

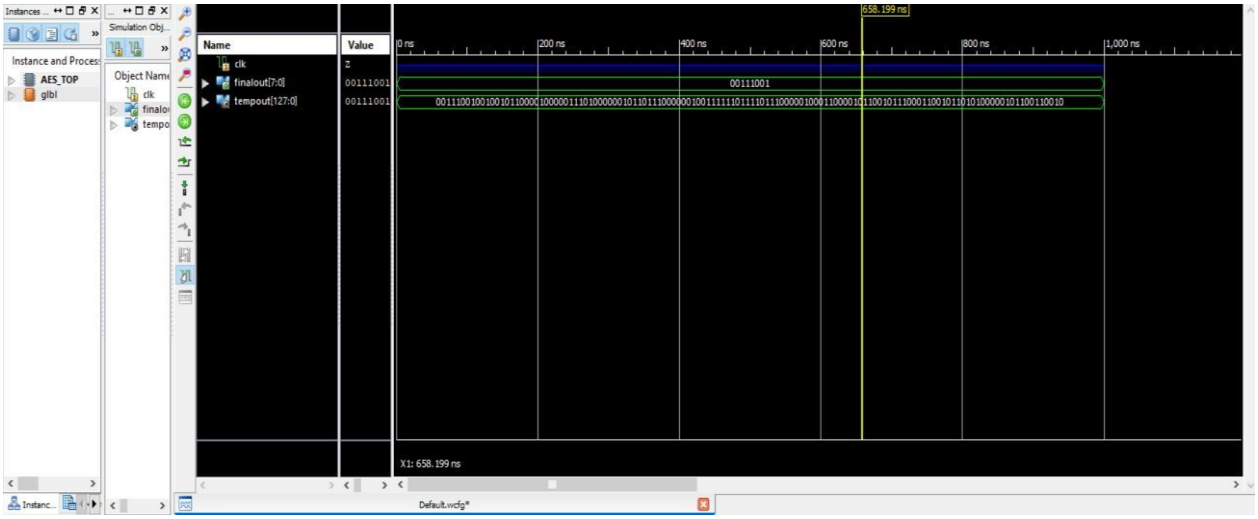


2)Rounds Operation



ISSN: 2233-7857 IJFGCN

3)Encrypted Data



V. CONCLUSIONS

This design presents a universal cryptography processor algorithm that supports both private and public key cryptography algorithms. We achieved this by expressing the primitives of three important algorithms for smart cards (DES, AES, and ECC) in terms of simple logical operations that maximize the number of common blocks among them. This approach resulted in a crypto-algorithm that meets performance specifications and time constraints. As switching between both the private key algorithms needs a good controllability. Also ECC algorithm is used for secure transfer between the communication entities.

. REFERENCES

[1] Data Encryption Standard (DES), Oct. 1999. Fed. Inf. Process. Standards Pub..
[2] Advanced Encryption Standard (AES), Nov. 2001. Fed. Inf. Process. Standards Pub..
[3] IEEE Standard Specifications for Public-Key Cryptography, Jan. 2000.
[4] S. S. Raghuram and C. Chakrabarti, “A programmable processor for cryptography,” in Proc. ISCAS, pp. V-685–V-688.
[5]M. Aydos, T. Yanik, and C. K. Koc, “High-speed implementation of an ECC-based wireless authentication protocol on an ARM microprocessor,” Proc. IEE Commun., vol. 148, no. 5, pp. 273–279, Oct. 2001
[6] J. Goodman and A. P. Chandrakasan, “An energy-efficient reconfigurablepublic-keycryptographyprocessor,”IEEEJ.Solid-StateCircuits, vol. 36, no. 11, pp. 1808–1820, Nov. 2001.
[7]E. Trichina, M. Bucci, D. De Seta, andR. Luzzi, “Supplemental cryptographic hardware for smart cards,” IEEE Micro, pp. 26–35, Nov.–Dec. 2001.
[8]P.H.W.Leong and I.K.H. Leung, “A microcoded elliptic curve processor using FPGA technology,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 10, no. 5, pp. 550–559, Oct. 2002
[9]I.Verbauwhede, P. Schaumont, and H. Kuo, “Design and performance testing of a 2.29 -GB/s rijndael processor,” IEEE J. Solid-State Circuits, vol. 38, no. 3, pp. 569–572, Mar. 2003.
[10]N. S. Kim, T. Mudge, and R.Brown, “A 2.3 Gb/s fully integrated and synthesizable aes rijndael core,” in Proc. IEEE Custom Integrated Circuits Conf., 2003, pp. 193–196.

ISSN: 2233-7857 IJFGCN