FPGA Implementation of Hybrid Cryptosystem

M. N. Praphul, K. R. Nataraj

Abstract—With the development of Computer Network and Communication Technology, a great mass of data and information need to be exchanged by public communication networks. High efficiency and high safety of data transmission become much more important. There are several information encryption algorithms of which, Advanced Encryption Standard (AES) and Rivest Shamir Adleman (RSA) are widely used two algorithms of symmetric encryption technology and asymmetric encryption technology respectively. The existing symmetric scheme-AES algorithm provides high speed stream for large data and uses less amount of computer resources but induces less degree of security in large amount of data. The asymmetric cryptographic algorithm or a public key cryptographic algorithm-RSA is more secure comparatively, as it has two keys one for encryption and another one for decryption, but is much slower and uses a huge amount of computer resources. In order to cope up with these short comings, a proposal to use an improved version of the hybrid encryption scheme is done, which is a combination of Advanced Encryption Standard (AES) and Rivest Shamir Adleman (RSA) with cross encrypted keys for secure key exchange and hybrid encryption for enhanced cipher-text security. This system is implemented on Spartan 3 FPGA using VHDL as the programing language.Synthesizing and implementation of the code is carried out on Xilinx -Project Navigator, ISE 12.1i suite.

Index Terms— Advanced Encryption Standard (AES), FPGA, hybrid encryption, Rivest Shamir Adleman (RSA),

I. INTRODUCTION

Cryptography is the art and science of achieving security by encoding messages to make them non-readable. The Cryptographic technique consists of encryption & decryption methods. These are the principal means to provide information security. Encryption method transform plain text message into cipher text, whereas decryption method transforms a cipher text message back into plain text. Not only has it to ensure the information confidential, but also provides digital signature, authentication, secret sub-storage, system security and other functions. Therefore, the encryption and decryption solution can ensure the confidentiality of information, as well as the integrity of information and certainty, to prevent information from tampering, forgery and counterfeiting. Encryption and decryption algorithm's security depends on the algorithm while the internal structure is the rigor of mathematics and also depends on the key confidentiality. Key in the encryption algorithm has a pivotal position, once the key is leaked, it means that anyone can be in the encryption system to encrypt and decrypt information; it means the encryption algorithm is useless.

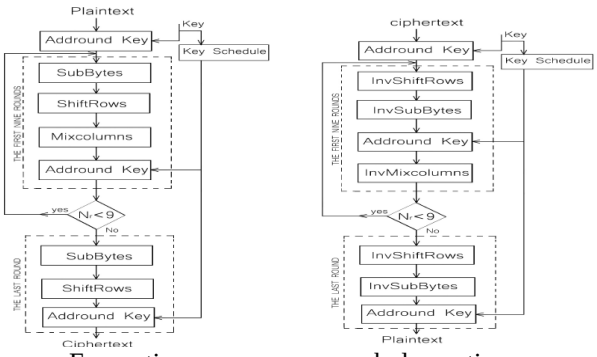
Therefore, what kind of data you choose to be a key, how to distribute the private key, and how to save both data transmission keys are very important issues in the encryption and decryption process.

There are several information encryption algorithms of which Advanced Encryption Standard (AES) and Rivets Shamir Adleman (RSA) are considered as the best two algorithms of symmetric encryption technology and asymmetric encryption technology respectively. The existing symmetric scheme-AES algorithm provides high speed stream for large Data and uses less amount of computer resources but induces less degree of security of data. In turn, the asymmetric cryptographic algorithm or a public key cryptographic algorithm-RSA is more secure, as it has two keys one for encryption and another one for decryption, but is much slower and uses a huge amount of computer resources. In this paper will develop an improved version of the hybrid encryption scheme, which is a combination of Advanced Encryption Standard (AES) and Rivest Shamir Adleman (RSA) with cross encrypted keys for secure key exchange and hybrid encryption for enhanced cipher-text security .The design will be tested on Xilinx Spartan 3 FPGA.

II. EXISTING AES and RSA CRYPTOGRAPHY ALGORITHM

The AES is a cryptographic algorithm that is used to encrypt (encipher), and decrypt, (decipher), information. Key Expansion generates a Key Schedule that is used in Cipher and Inverse Cipher procedures. Cipher and Inverse Cipher are composed of specific number of rounds (Table 1). For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key length. Table I Comparison of block size, key length and number of rounds in AES

Type	Block Size Nbwords	Key Length Nkwords	Number of Rounds Nr
AES -128 bits key	4	4	10
AES -192 bits key	4	6	12
AES -256 bits key	4	8	14



a. Encryption b.decryption

Fig. 1 block diagram of AES algorithm.

Manuscript received June, 2013.

M.N Praphul. ECE, SJBIT, Bangalore, India.
Dr K.R.Nataraj,HOD,ECE,SJBIT,Bangalore,India.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A. Aes operation

Following indicates the transformation in AES algorithm based on the structure in Fig 1.The brief introduction is listed as below**:**

1. The SubBytes operation: The SubBytes operation is a non-linear byte substitution, operating on each byte of the state independently [8]. The substitution table (S-Box) is invertible and is constructed by the composition of two transformations: Take the multiplicative inverse in Rijndael's finite field .Apply an affine transformation which is documented in the Rijndael documentation. Since the S-Box is independent of any input, pre-calculated forms are used. Each byte of the state is then substituted by the value in the S-Box whose index corresponds to the value in the state is a(i,j) = SBox[a(i,j)] .

2. Shift row transform: Cyclically shifts the rows of the State over different offsets[4]. The operation is almost the same in the decryption process except for the fact that the shifting offsets have different values. The goal of this transformation is to scramble the byte order inside each 128-bit block.

3 .Mix column transform: This process is for mixing up of the bytes in each column separately during the forward process. The corresponding transformation during decryption is denoted Inv Mix Columns and stands for inverse mix column transformation. The goal is here is to further scramble up the 128 –bit input block.

4. *Add round key and key expansion:* In this operation, the round key is applied to the State by simple bit by bit XOR. Basically Key Expansion unit is used to generate the next round key as for three different key size, AES consist of 10, 12 or 14 rounds. So after every round a new round key need to be produced. So this unit produces that round key for each round. This unit also utilizes the concept of shifting the bytes and substitution of bytes which were nused in data processing unit.

5. Key Schedule: Key scheduling is a critical process in AES that generates (Nr+1) round keys based on an external single key. The Key expansion process of AES algorithm uses a Cipher Key K to generate a key schedule. This generates Nb(Nr+1) words, of which the algorithm requires initial Nb words and each of the Nr rounds, require Nb words of Key Data. Key scheduling can produce keys either on the fly or store them in an internal key memory the key setup phase and then read them from this memory whenever required by the encryption/decryption unit. The critical path of Key Expansion is shorter than that of any round, speed of the system can't be enhanced by reducing the critical path of Key expansion. Keys on the fly eliminate the requirement for key storage, but brings overhead for decryption since decryption begins after the last roundkey is generated.

B.The RSA Algorithm

The RSA algorithm is used for both public key encryption and digital signatures. It is the most widely used public key encryption algorithm [6]. The basis of the security of the RSA algorithm is that it is mathematically infeasible to factor sufficiently large integers.

1. Key Generation Algorithm

The RSA Cryptosystem requires the use of a public key and a private key. Both these keys must fulfill certain conditions to ensure the integrity of the system. The following steps illustrate the key generation**:**

Choose two large prime numbers of approximately the same size, namely p and q.

i. Compute the product of these two primes, n = pq.
ii. Also, compute the value of φ(n) = (p-1)(q-1).
iii. Choose an integer e between 1 and φ (n) such that gcd (e, φ (n)) = 1.
iv. Finally, compute d whereby d = e^{-1} mod (φ (n))**.**

The public key is (n, e) whereas the private key is (p, q, and d)**.**

2. Encryption and Decryption

When Bob intends to send an encrypted message to Alice, these are the steps to be taken**:**

i. Obtain Alice's public-key (n,e), which should be listed in a public directory.
ii. Represent the plaintext message as a positive integer x, whereby x < n.
iii. Compute the ciphertext using the encryption function: y = $e_K(x) = x^e$ mod n
iv. Transmit the ciphertext to Alice.

Upon receiving the encrypted message, there are several steps to be taken by Alice:

i. Compute the integer representation of the plaintext using the decryption function: x = $d_K(y) = y^d$ mod n and her own private key (p,q,d).
ii. Decode the corresponding plaintext from its integer representation, x.

III. PROPOSED IMPROVED HYBRID CRYPTOSYSTEM

This paper proposes a hybrid cryptosystem that utilizes benefits of both symmetric key and public key cryptographic methods. Symmetric key algorithm (aes) is used in the crypto system to perform data encryption and decryption. public key algorithm (rsa) is used in the crypto system to provide key encryption before key exchange. Combining both the symmetric-key and public-key algorithms provides greater security and some unique features which are only possible in the hybrid system shown in Fig 2. The implementation has various modules of aes and rsa.

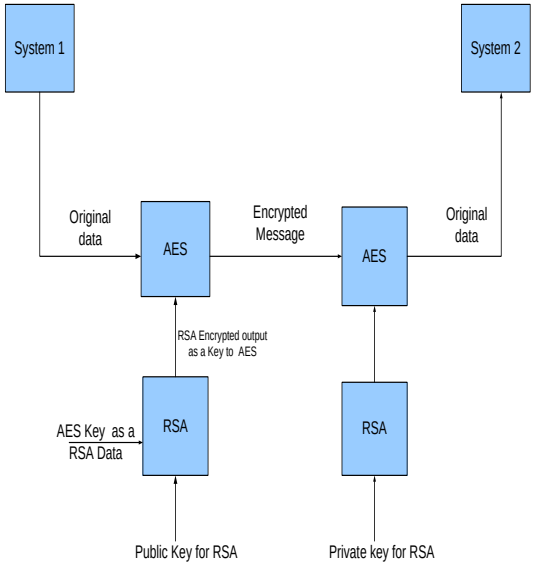


Fig 2. Proposed hybrid cryptosystem

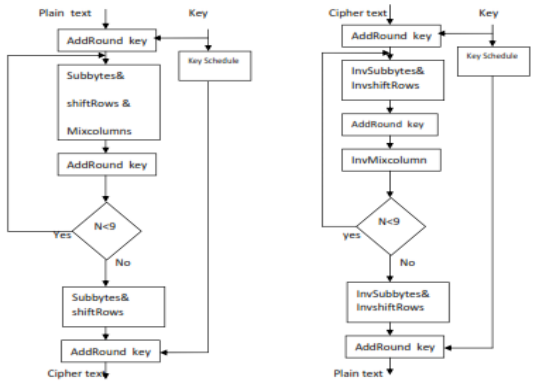
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

A. Modified block description

The original data/message is sent for encryption using AES.The key used for the AES algorithm is further encrypted as a RSA data by the RSA algorithm using recipient's public key. Both the encrypted secret key and the encrypted message are then sent to the recipient. The recipient decrypts the secret key first, using its own private key, and then uses that key to decrypt the message.

1. Optimised AES Algorithm used in proposed hybrid cryptosystem

In standard AES algorithm, there are four steps like SubByte, ShiftRow, MixColumn and Add Round Key in normal rounds. Our design highlights some following modification, Exclusion of Shift Row is performed through calling required shifted element from the data matrix, (instead of calling element one by one sequentially orderly from the data matrix); thus merging of the two steps SUB-BYTE and SHIFT ROW reduces one step. Further merged sub bytes and shift rows is combined with mix columns to form T box as shown in Fig 3**.**



a. Encryption b.Decryption

Fig 3 Optimized AES algorithm to be used in Hybrid cryptosystem

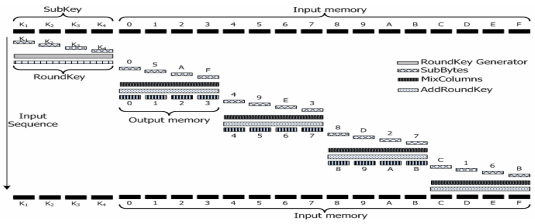


Fig. 4. The internal data flow of the optimized AES architecture

In the proposed design, the entire round operation consists of five phases [1]. Each phase executes the four basic transformations in sequence, as shown in Fig. 4. The first phase is executed to obtain the round key. Each of the remaining four phases performs the four basic transformations for the input block.

Stage 1: Read K1 from the Subkey, execute SubBytes, and write the result to the corresponding locations in the RoundKey Generator. Repeat for K2, K3, and K4. Then generate the round key from the result obtained from Round key generator

• *Stage 2*: Read input byte 0 from the input memory. Execute SubBytes and write the result to the corresponding location in MixColumns. Repeat for 5, A, and F, then run MixColumns for the written 4 bytes, and write the result to the output memory addresses 0, 1, 2, and 3, respectively.

• *Stage 3*: Repeat the second phase for input bytes 4, 9, E, and 3 in the same way. Write the result to the output memory addresses 4, 5, 6, and 7, respectively.

• *Stage 4*: Repeat the second phase for input bytes 8, D, 2, and 7 in the same way. Write the result to the output memory addresses 8, 9, A, and B, respectively.

• *Stage 5*: Repeat the second phase for input bytes C, 1, 6, and B in the same way. Write the result to the input memory addresses C, D, E, and F, respectively. The input memory C, D, E, and F are reused as the corresponding outputs for the next round.

2. Modified RSA algorithm used in proposed hybrid system

i. Choosing the Modulus for the RSA Algorithm

With the definitions of d and e as presented earlier, the modulus m must be selected in such a manner that the following is guaranteed:

$$(P^e)^d \equiv P^{ed} \equiv P \pmod m \qquad (1)$$

Since, $C = P^e \pmod m$**,** is the encrypted form of the message integer M and decryption is carried out by $P=C^d \pmod m$.It was shown by Rivest, Shamir, and Adleman that, n is a product of two prime numbers: $n = p \times q$ for some prime p and prime q[6].

ii. Choosing a Value for the Public Key Exponent e and N private key d.

Encryption consists of raising the message integer M to the power of the public exponent e modulo n. This step is referred to as modular exponentiation. The mathematical requirement one is

$$\gcd(e,\varphi(n)) \qquad (2)$$

since otherwise multiplicative inverse mod wouldn't exist φ (n).Since $n = p \times q$, this requirement is equivalent to the two requirements

$$\gcd(e, \varphi(p)) = 1 \text{ and } \gcd(e, \varphi(q)) = 1. \qquad (3)$$

$$\gcd(e, p-1) = 1 \text{ and } \gcd(e, q-1) = 1. \qquad (4)$$

Once a value for the public encryption exponent e is found, the next step is to calculate the private decryption exponent d from e and the modulus n. $d = e^{-1} \bmod \varphi(n)$.Calculating $e^{-1} \bmod \varphi(n)$' is referred to as modular inversion.

IV. ARCHITECTURAL DESIGN IMPLEMENTATION

The proposed work is implemented using the Xilinx ISE 12.1(version) and Xilinx Spartan 3E FPGA prototyping board has been used for the hardware implementation and testing, which is illustrated in Fig 5 .VHDL is used as the hardware description language because of the flexibility to exchange among environments. The software used for this work is Xilinx ISE 12.1.

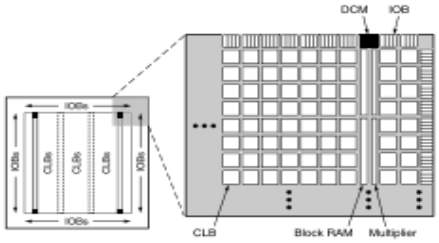


Fig 5. Spartan-3 Family Architecture.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

This is used for writing, debugging and optimizing efforts, and also for fitting, simulating and checking the performance results using the Xilinx xst simulation tools available on Web pack design software. An iterative method of design is to implement to minimize the hardware utilization. The hybrid encryption and decryption process module schematic is shown in figure 5. In order to allow a full parallel process of the state, it is necessary to implement all the transformations over 128 bits. The most expensive one is the using private key generated by RSA for to encrypt AES key.

1. Implementation of sub bytes/inverse sub bytes(s box, s^{-1} box)

The S-boxes are too complex to be implemented directly so look-up tables (LUT), initially loaded with the correct values are used. Each of them takes 8 bit and gives 8 bit back, we need four 8x8bit ROM [8]. The 8bit input is divided in 3 parts: the bits 0,1,2,3 are used as an address for 16 bit ROMs, the bits 4, 5, 6 are used to control which ROM is controlling the bus, and the last bit chooses which bus will be read. Each S-Box is 9x9 CBLs. The main part is 8x8 CLBs: each CLB contains two 16x1 ROMs, connected to a bus by a TBuf as shown in Fig 6.

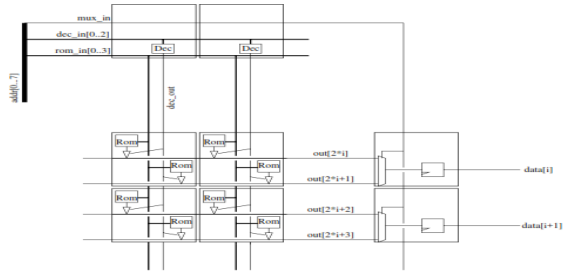


Fig 6. ROM 8X8 used for S-box implementation

2. Implementation of ShiftRows.

ShiftRow shifts the rows of the block by 0,1,2,3, so all the columns of one block gets the resulting column. Two blocks are computed serially: what comes in ShiftRow is written in one block, what goes out is read from the other one, and the two blocks are exchanged every four cycles.RAM of 8x4 words are needed as shown in Fig 7, i.e. a 8x32 bits, with four distinct addresses for writing and four other for reading (only one read-address is needed for the encoding, but for the decoding, four read-addresses and one write address are needed. CLB is used as RAMD; each CLB is used as a 16x1 bit RAM. Addresses are coded on four bits, the last one being set to zero.

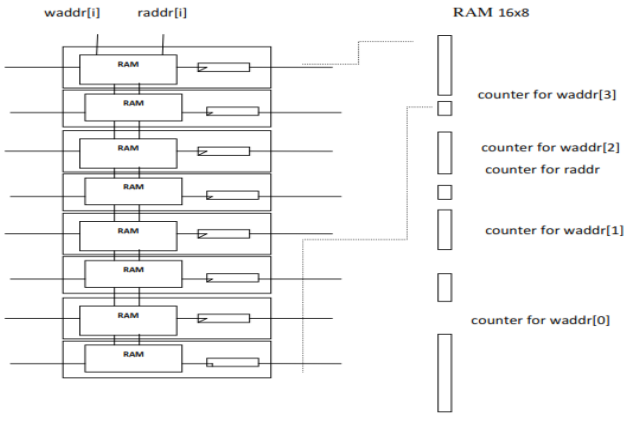


Fig 7: RAM 8x32 used in shiftrows

The size of this structure is 4 blocks of 1x8 CLBs. Four counters mod 4 and one counter mod 8 in the same column is placed, in the 8 CLBs left: for the encoding, the only read-address is given by a counter mod 4, plus one bit that indicates which block data is read in, and the four write-addresses are given by four shifted counters mod 4, the last bit being the opposite of the read-address one.

For the decoder, either read and write addresses are exchanged, or write in diagonal the other way (by setting the initial values of the counter to their opposite).

3. Implementation of T- Box / Inverse T-Box .

The subbytes, shiftrows and mixcolumns transformations are combined and can be implemented as a whole. This is referred as T-box approach box approach for encryption process only.

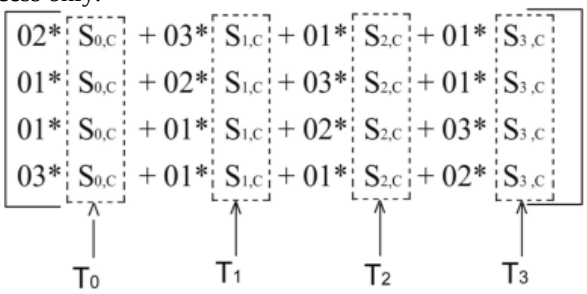


Fig 8: T-Box Tables for Encryption.

The resultant matrix shown above has each column with fixed constants multiplied by state inputs. These state inputs are updated with the next row elements for the next iteration. That is, first row elements of the state inputs are always multiplied with the first column constants of the multiplication matrix. Similarly, the second row elements of the state inputs are always multiplied with the second column constants of the multiplication matrix and so on. The first row of the state input uses table T0, the second row uses the table T1, the third row uses the table T_2 and fourth row uses the table T_4.

4 .Optimised rsa algorithm

Montgomery Modular Multiplication (radix 2) is efficient for computing P*C mod m, where

$C=\sum_{i=0}^{n+1} c_i 2^i,\ c_i \in \{0,1\}, c_0=0.$ (5)

$P=\sum_{i=0}^{n+2} a_i 2^i,\ a_i \in \{0,1\},\ a_{n+1}=0; an_{+2}=0.$ (6)

$R_0=0$;For i=o to n+2 do $q_i=R_i(0)$ (7)

$R_i+1=(R_i+a_i .C + q_i . M)/2$ (8)

The loop above is executed more number of times. This measure simplifies the computation of q compared to the original algorithm. With this step ,it is sure that inequalities $R_i<3m$ and R $R_{n+3}<2m$ always hold. The result of a modular multiplication R_{n+3} can thus be reused as input P and C for the next multiplication. The originally proposed final comparison and subtraction is avoided and a pipelined execution of the algorithm possible.

RSA encryption and decryption computation system consists of Processing elements to compute MPWID bits of a modular multiplication.

In the processing elements we need the following main registers of MPWID bit size.

- modreg(MPWID): store the modulus value during operation.
- mpreg(MPWID-1): storage of the multiplier values.
- mcreg(MPWID+1): control of the clock enable.

- prodreg(MPWID+1) : storage of the result at the end of a multiplication.

5. Methodology

In our implementation we adopted the following design flow approach that resulted in fast verification of gate level netlists:

i. Design entry
ii. Logic verification
iii. Synthesis
iv. Place and Route
v. Timing Verification

The entire design, with the exception of vendor specific soft macros, was entered in VHDL format. Once the design is developed in VHDL, boolean logic and major timing errors were verified by simulating the gate level description with model sim and .The next step involved the synthesis of the VHDL code with xilinx synthesis tool. The output of this step was an optimized netlist describing the gate level degn in XILINX format. The most time consuming step was the compilation of the synthesized design with the place and route tools available from Xilinx.The final step of the design flow was to verify the design once again but this time with the physical net, CLB, and pad delays introduced when the design was placed into specific device. This was accomplished with the same testbenches and simulation models that were used during the logic verification stage.

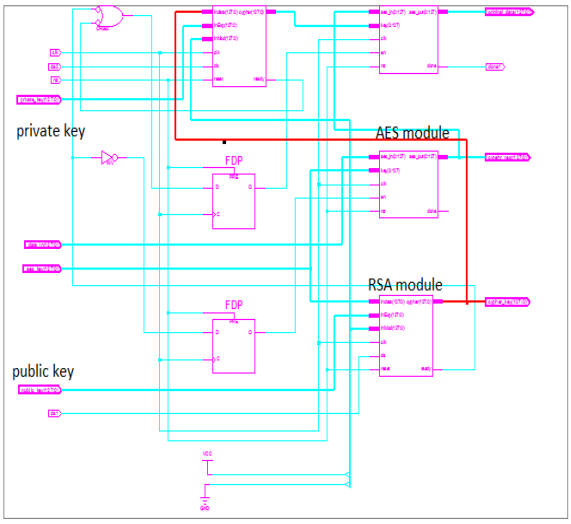


Fig. 9 Schematic diagram of Hybrid Encryption Process

V. SIMULATION RESULTS

ModelSim SE 6.3 software is used for simulation and optimization of the synthesizable VHDL code. Synthesizing and implementation (i.e. Translate, Map and Place and Route) of the code is carried out on Xilinx - Project Navigator, ISE 12.1i suite. The schematic design is shown in Fig 9.Initially hybrid system is designed in such a way that it should be implemented on a single chip, but when implemented on FPGA kit SPARTAN-III[11], the design exceeded more than 6 million gates. This made impossible to implement the hybrid design module to dump on a single chip. As a result hybrid Encryption module is implemented on a FPGA for the verification of the hardware implementation of the design. The output of encryption is fed to the decryption module and observed that the output has regenerated the original text at the output in Fig 8. An input is forced to the design and the original data is obtained back after the decryption of the encrypted message with the use of different key.

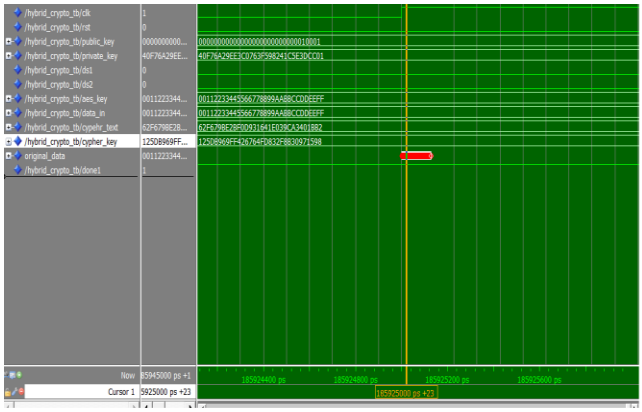


Fig 10: Simulation of 128-bit hybrid Encryption and Decryption.

Table II shows the number of Configurable Logic Block (CLB) occupied blocks (I/OB) and the clock frequency for the implementation of the architecture of Hybrid cryptosystem.

TABLE II: Resources utilization of Hybrid cryptosystem

Logic Utilization	used	available	utilized
Number of Slices	1269	3584	35%
Number of Slice Flip Flops	2235	7168	31%
Number of 4 input LUTs	495	7168	6%
Number of bonded IOBs	68	141	48%
Number of GCLKs	1	8	12%

Optimal architecture that permits to use 3589 CLBs (35%) and 48% Input/Output Block of this circuit with a clock frequency of 87.704 MHz is used.

TABLE III: Hybrid Encryption and Decryption

Process	**Clock** frequency(MHz)	Delay (ns)
Encryption	87.704	10.402
Decryption	87.704	11.23

We implemented our design for following optimised algorithm key bit lengths(K).Table 1V shows our results in terms of used CLBs (C),frequency F,clock cycle time (T) and the time–area product (TA).

TABLE IV :AES and RSA comparation

Algorithm	K	F (Mhz)	T (ns)	C	TA (C.ns)
AES	128	65	16.5	1207	19915.5
RSA	128	65	19.8	1122	22215.6

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

VI. CONCLUSION & FUTURE WORK

This work proposed a hybrid cryptosystem that utilizes benefits of both symmetric key and asymmetric cryptographic methods .The encryption and decryption of any data has a secure key, which is used for data encryption and decryption. For this purpose asymmetric key is used. One of the approaches is to generate a random secret key of 128 bits for a symmetric cipher-AES, and then encrypt this key via an asymmetric cipher-RSA, using the recipient's public key of 128 bits. The message itself is then encrypted using the symmetric cipher and the secret key. Both the encrypted secret key and the encrypted message are then sent to the recipient. The recipient decrypts the secret key first, using his/her own private key,and then uses that key to decrypt the message .Thus providing higher degree of security to data transmission. Further step is to study the behavior of the Hybrid cryptosystem model in various input conditions like speech,video signals and images as an input and to implement as an ASIC. There is a provision and flexibility to remove or add any other cryptographic standards in the proposed system.

ACKNOWLEDGMENT

With profound sincerity and gratitude, I acknowledge my constant discussion with our HOD **Dr.K.R.Nataraj.**, for his valuable guidance and support extended during the work.

REFERENCES

1. Ohyoung Son and Jiho Kim "Compact Design of the Advanced Encryption Standard Algorithm for IEEE 802.15.4 Devices "Journal of Electrical Engineering & Technology Vol. 6, 2013,
2. Avi Kak , Avinash Kak, "Computer and Network Security on Public-Key Cryptography and RSA" May 15, 2013 Purdue University
3. N.Singh, G .Raj. "Security on bccp trough AES encryption technique",Special Issue of INTERNATIONAL journal of engineering science & advanced technology (2250–3676) Jul-Aug.2012.
4. Jeneba mary.B "hybrid cryptography by the implementation of rsa and aes"international Journal of Current Research, Vol. 3, Issue, April, 2011
5. Alan Daly and William Zhenzhen Liu. "Implementation of AES Encryption based on FPGA". Modern electronic technology.
6. Marnane "Efficient Architectures for implementing Montgomery Modular Multiplication and RSA Modular Exponentiation on Reconfigurable Logic". -University College Cork Ireland 2010.
7. Yu; Tong Li; Na Zhao; Fei Dai "Design and implementation of an improved RSA algorithm", April 2010
8. Song J. Park "Analysis of AES Hardware Implementations" Department of Electrical & Computer Engineering Oregon State UniversitCorvallis,
9. Tim Good and Mohammed Benaissa. "AES on FPGA from the Fastest
10. to the Smallest".
11. Panu Hämäläinen, Marko Hännikäinen, Timo Hämäläinen, and Jukka Saarinen. "Hardware implementation of the improved wep and rc4 encryption algorithms for wireless terminals" ,2010
12. Tim Güneysu. J Cryptogram Eng "Utilizing hard cores of modern FPGA devices for high-performance cryptography". 2011
13. Benjamin Leperchey, Charles Hymans "FPGA implementation of the Rijndael algorithm" June 9, 2009
14. Shanxin Qu,Guochu Shou,Yihong Hu,Zhigang Guo,Zongjue Qian. "High Throughput Pipelined Implementation of AES on FPGA". International Symposium on Information Engineering and Electronic Commerce.2009
15. Behrouz A.Forouzan "Cryptography and network security "TATA Mcgraw hill publication 2007 edition.

AUTHORS PROFILE

M.N. Praphul. obtained B.E (Electronics & communication) from SDM institute of technology,Ujjire in the year 2011.He is currently in Mtech (Vlsi & Embedded systems) at SJB institute of technology.He has presented a paper in National level conference communication and computation.

Dr. K .R. Nataraj, is Prof & hod E&C Department, SJB institute of Technology,Bangalore. He obtained B.E (Electronics & communication) from Siddaganga Institute of Technology Tumkur in the year 1995. He obtained M.E (power electronics) from BMS college of engineering Bangalore in the year 2000. He obtained PhD in Dr MGR Deemed University, Chennai under the guidance of Dr B.S.Nagabushana PhD (IISce) Director SAN Lab Technologies Bangalore and Dr S Ramchandran PhD (IIT Madras) in the year August 2010. His current research area is vlsi designing. He has 7 International Journals published.