International Journal of Engineering & Technology, 7 (2.31) (2018) 219-224

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Implementation of various data encryption methods for medical information transmission

S. Neelima[1*], R. Brinda[2]

[1]Gandhiji Institute of Science and Technology, Jaggayyapet. Krishna District, Andra Pradesh.
[2]HOD of ECE, Avinashilingam Institute for Home Science & Higher Education for Women, Faculty of Engineering, Coimbatore, Tamilnadu.
**Corresponding author E-mail:seethalaneelimaph.d@gmail.com*

Abstract

Encryption is the process of converting the data from readable format into unreadable format with help of any mathematical expression or sometimes with the help of key. On the other hand decryption is the reverse process of encryption with help of same key used at encryption or with the help of some other key. The paper presents the different methodology used for encryption and decryption. Several methods presented in the literature are reviewed. The methods- Rivest-Shamir-Adlemen algorithm, Data Encryption Standard, Advanced Encryption Standard and three different Secure Hash Algorithm are reviewed and implemented using various FPGA devices. The power consumption, delay and area are analyzed and compared. From the analyses it is been found that the performance of AES and SHA3 are better when compared to other algorithms. These algorithms provide high security when compared to rest of the methods.

Keywords: *RSA, AES, DES, AES, SHA, encryption, decryption, FPGA, power consumption, delay, area*

1. Introduction

The data communication between different electronic sources and receivers through different medium has made the data security becomes more important. But in some cases the data communication needs data security. Because there is a possibility to unauthorized access of electronic data. To prevent data from those accesses they need security system. To ensure that there is many security algorithms are available. While providing data security the cryptography is the most important term. It provides privilege to store or transmit the data through any in secured medium. Cryptography is nothing but a cluster or many algorithms which provides data security. It ensures the data security through convert the data into some prescribed format. This process is called ciphering. Then the resulting data will becomes readable only by indented recipient. The data cannot be modified by any other unauthorized person. The process of converting the data into non readable format is known as ciphering or enciphering or encryption. The reverse process of ciphering is called deciphering or decryption.

Based on usage of key the cryptography can be classified into two types those are private key cryptography and public key cryptography. The private key cryptography means that, they use same key for both encryption and decryption. On the other hand the public key cryptography uses separate key for encryption and decryption. The best example for public key encryption is RSA (Rivest, Shamir, and Adleman) algorithm. It is based on the principle of number theory. In RSA algorithm same process (mathematical process) is followed in both encryption and decryption but with the different keys. That's why it becomes the example for public key cryptography. The main concern is key size. Because the security level purely depends on the key length.

The DES (Data Encryption Standard) is an example for private key encryption. It uses same key for both encryption and decryption. Due to the key length it becomes broken very easily. It has the key length of 56bits. To overcome this scenario NIST announced some other algorithms which provide high security. One among them is AES (Advanced Encryption Standard). It provides efficient data security than the previously discussed algorithms. The AES algorithm provides data security in many applications like cellular networks, web servers, mobile networks, smart cards etc. on the other hand most of financial transactions are becomes online. It provides data security to money exchange also. This algorithm achieves good robustness because it utilizes both hardware and software for implementation. The key length and number of rounds additional strength. On the other hand it becomes mathematically complex due to more number of rounds.

To reduce mathematical complexity, the demand on hash functions increased logarithmically, because it generates fixed length hash functions even for variable length data. The fixed length hash functions are very easy to check with input data. The specific property of hash function is that it provides collision resistance and confidentiality. Hash functions are applicable for digital signatures, message authenticity, and password protection. Based on required output block size there is many hashing algorithms are available. They are also called hashing codes. The block sizes are varying from 128 bits to 512 bits. Based on the input block size the hashing algorithm consist number of rounds. Each round is provided with two different inputs. A round has the most recent input block size and output of previous round. Also each round is provided auto generated key. For the key generation process the system consist of separate key generation block and hardware module. The output of previous round is given to input of next round which alter the current stage and the same procedure is followed for the next consecutive rounds and so on. The same process is followed for number of rounds. The confidential level of digital data increases based on the more number of rounds.

The process of altering the data in second round by the first round is known as avalanche effect of hash function. It shows that the hash function will provide two different outputs for two inputs

they differ by a single bit. The hash functions having different types based on its structure, such as SHA-0, SHA-1, SHA-2 and SHA-3. SHA-1 is suggested by NIST in 1993, due to some security vulnerabilities it will not become more familiar. To overcome those, in 1995 SHA-1 algorithm is proposed. It most used algorithm in SHA family. SHA-1 is efficient than SHA-0, even though SHA-2 algorithm is proposed in 2005 to offer security for practical data and long time employability. SHA-2 algorithm is further classified into four types, such as SHA-224, SHA-256, SHa-384 and SHA-512. SHA-2 also an efficient algorithm. But in 2012 NIST announced keccak algorithm as SHA-3. It offers many facilities like robustness, resistance from vulnerable attacks.

2. Literature survey

A. RSA- Rivest, Shamir and Adlemen algorithm

As the major problems in most communication systems is the secure transportation of data, new findings are required to meet the needs. Several algorithms are proposed. Encryption algorithm implementation through several means from software [**Adriana Naydenova Borodzhieva, Sangita A. Jaju, Santosh S. Chowhan]** to hardware help the communication sytem more secure. Various FPGA implementation was carried out in literature for RSA which has high complexity and resource requirement issues [**A.R.Landge, A.H. Ansari]**. A hardware based implementation of RSA algorithm in FPGA was presented by **Amit Thobbi et al [].** The length of the cipher text is 64 and the public key was 128bit in the RSA algorithm. The coding was done in VHDL using Xilinx. Speed and power optimization was carried out. The tradeoff between space power and speed was presented. On the other hand **Muhammad I. Ibrahimy et al** developed the FPGA with flexible key RSA encryption standard. Modeled in VHDL the RSA encryption engine supports multiple key sizes with different levels of security. RSA operation was tested with nested loop addition and subtraction and processing time and amount of space in the FPGA was observed. Altera STRATIX II device was used. The RSA algorithm suffers from side channel analyses like timing attack, fault induction attack, simple/differential power analysis and electromagnetic analysis. The Montgomery Multiplication based on Residue Number Systems with RSA signature in parallel architecture was implemented in FPGA [**Mathieu Ciet]**. Electromagnetic analysis was introduced in the literature. But the method suffers from high memory requirements and large delay even though the security level was higher. The Montgomery multiplication based method been extended till 1024 with high speed FPGA kit [**Velibor Skobic** et al].

Few methods rely on shift and add operation for the implementation of various blocks of the RSA algorithm [**Sushanta Kumar Sahu, Manoranjan Pradhan]**. The area occupied was less with different key support from 128 to 512bits when implemented by VHDL in FPGA kit. The method was having medium speed profiles but the security level is less. Still key generation using random number generator LFSR was familier [**Rohith S, Poornima, Mahesh C].** For prime number detection "Sieve of Eratosthenes" algorithm was done. The various processing elements can be utilized to increase the efficiency. Especially Booth multiplication and Extended Euclidean algorithm to find GCD of the public key are dominant. Similar to previous literatures the Modular Multiplication and Modular Exponentiation by using LR binary method increases the security level for the encryption and decryption.

B. DES encryption algorithm

In the series of encryption standards evolved the DES algorithm based encryption was efficient. The algorithm suffers from weaker relativity between the generation of sub-key and the critical arithmetic. Adopting preferential resources in reconfigurable device shows improved performance [**Ji Yao and Hongbo Kang].** Speed can be increased using pipelined methods.

The independent implementations of round-function and key generator reduce the logic complication of adjacent pipeline. The randomness of the encryption key can be enhanced using irrational Numbers **[Jing Wang et al].** This increases the speed threshold to control permutation. As hybrid methods are always a welcomed one in research DES and RSA combined in Bluetooth was presented by **Wuling Ren and Zhiqian Miao (2010).** To avoid the intruders in usiual 128 bit symmetric stream ciphered Bluetooth data transmission, a hybrid method instead of the E0 encryption, DES algorithm is used for data transmission because of its higher efficiency. block encryption, and RSA algorithm is used for the encryption of the key of the DES.

C.AES algorithm

AES the Rijindael's encryption and decryption standard was a effective alternative for the existing methods. It is the most popular symmetric cryptographic algorithm utilized in secure data communications like ebanking. Optimizing and implementing the same in software and hardware are always a challenging task [**Guang-liang Guo et al (2015)].** Modifications can improve the speed and performances. The optimization can be done using permutation data scrambling approach **[Dilna and Babu (2016)].** The permutation increases the area efficiency even though it resembles to Data Encryption Standard (DES) algorithm. AES used here have 128 bit plaintext and keys which converted into four 32bit blocks and exclusion of shift row. But high potential usage of same cryptographic algorithms risks the secure data processing so advanced changes should be incorporated. The longer the encryption keys the higher will be the security. For a 128bit key the throughput can be made so higher [**Nacci et al, 2014].** The other improvement would be the replacement of the substitution step in Advanced Encryption Standard (AES) algorithm. The substitutions are carried out using algebraic method through which the key sequence are controlled [**Luminiţa Scripcariu and Petre Daniel Mătăsaru, 2013]. A** Galois Field mathematic function with 8-bit elements dynamically controls the static box. Implementation of the AES algorithm is carried in various devices like embedded processors, DSP processors, Matlab and FPGA devices. **Pradnya Katkade, Dr. Mrs G.M Phade** implemented the AES algorithm in NIOS II processor and compared with FPGA which was implemented by several authors **[Pritamkumar et al 2005; J. Senthil Kumar, 2014;Mazen EI Maraghi; Junfeng Chu and Mohammed Benaissa, 2012]**

D. SHA methods

Magnus Sundal and Ricardo Chaves, 2017 presented the FPGA implementations of the SHA-3 hash functions. The architecture was improved through unfolded and pipelined structures. The area was reduced by solving the intra-round dependencies and memory mapping. The area reduction reduces the delay. **Muhammad Arsalan et al 2013** X design of Blake-256 algorithm is implemented on FPGA. Horizontal Folding and pipelining technique is used in which two Half-G functions are used to execute overall round function. Distributed Block Memory is used for storing permutation table values. A similar work was proposed by **Fatma Kahri, 2013** using Blake 256, **Kris Gaj et al and Jens-Peter et al**. **Kuntoro Adi Nugroho et al 2016,** proposed Secure Hash Algorithm (SHA) 2 and 3 and evaluated the entropy analysis and running time analysis.

The evaluation was carious out for a question shuffling problem for computer based test in which the randomness and computational efficiency was measured. For compact implementations Grøstl and Keccak provides better efficiency [**Bernhard Jungk and J¨urgen Apfelbeck 2011).** But for multimessage hashing, to obtain high security the hardware

architecture has to be optimized **(Yusuke Ayuzawa et al and Bernhard Jungk, Marc Stöttinger).**

3. Background methodology

A.RSA algorithm

RSA stands for Rivest, Shamir, and Adleman who invented the RSA algorithm on 1978. This algorithm uses the public key for encryption and private key decryption. The encryption key can be known by everyone. But the decryption key can known only by the receiver. That's why this method is more confidential. The receiver need to confirm that the received message can be send by an authorized sender or not. For that both encryption and decryption key send along with the encrypted message to the receiver. This is known as digital signature (Amit Thobbi et al, 2015: A.R.Landge and A.H. Ansari, 2013: Muhammad I. Ibrahimy et al, 2007; Adriana Naydenova and Borodzhieva, 2016). By comparing those signature receiver can realize about sender. The flow diagram of RSA algorithm is shown in Figure.1
The RSA algorithm performs encryption on sender side with the help of public key. Also performs decryption on receiver side by using private key, which is provided by the sender.

Finding prime number

In RSA algorithm first step is that finding the prime numbers such as P and Q. The value of n is calculated from the chosen prime numbers P and Q. n is derived by multiplying the both P and Q. The value of P and Q should be chosen carefully such a way that the value of n should not be identified by the third person. Then the resulting prime number becomes more confidential.

$$n = p * q \quad (1)$$

Calculation of totient's function (φ (n))

φ (n) is named as totient's function. It is calculated from P and Q. The totient's function used further to calculate the encryption key (e). φ (n) is calculated as follows

$$\varphi(n) = (p-1)*(q-1) \quad (2)$$

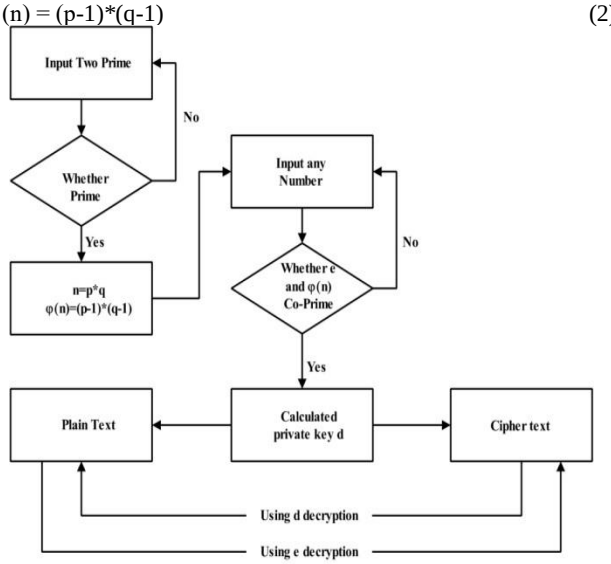


Figure 1: Flowchart of the RSA algorithm

Finding the public key –'e' (for encryption alone)

To calculate the encryption key, that should satisfy two conditions are as follows. The value of 'e' is acceptable only when it satisfies following two conditions.

$$1 < e < \varphi(n)$$
$$GCD(e, \varphi(n)) = 1$$

The resulting value of 'e' after satisfying above mentioned condition it will be considered as public key (for encryption).

Determination of private key -d

The private key'd' is multiplicative inverse of public key 'e'. The relation between private and public key is based on controversial concept. The controversy relation is expressed as

$$e * d = 1 + \text{mod } \varphi(n) \quad (3)$$

The private key (d) is calculated from the controversy concept is follows

$$d = [1 + k\,\varphi(n)]/e \quad (4)$$

The resulting integer 'd' is known as private key for decryption

Encryption method

Once the key for both encryption and decryption is identified then the encryption is performed. To perform encryptions consider the scenario where the plain text (which is to be encrypted) is M, the resulting text after the encryption (cipher text) is C. Encryption is performed by following expression

$$C = M^e \bmod (n) \quad (5)$$

Where
e is public key for encryption
n is prime number

Decryption method

The private key (d) is used here to perform decryption process. The resulting text after decryption is plain text. i.e. message. The decryption is performed through following expression

$$M = C^d \bmod n \quad (6)$$

B. AES (Advanced Encryption Standard)

AES (Advanced Encryption Standard) is the algorithm for the encryption of electronic data from the various sources. The algorithm converts the data into encrypted format with help of key; the encrypted data cannot be read by third person. The receiver can read the encrypted data after the decryption is performed. For both encryption and decryption the algorithm uses same key. That's why this algorithm also known as symmetric key algorithm. The steps followed in the algorithm are given below,

	Algorithm steps for AES Encryption standard
Initial round	Add round key
First round	Sub bytes
	Shift rows
	Mix column
	Add round key
Final round (no mix columns)	Sub bytes
	Shift rows
	Add round key

In the process of encryption and decryption the algorithm follows same steps repeatedly. Each step is considered as round. The number of rounds depends on the block size of input data to be modified. The size input block varies based on the user requirement. For example AES-128 consists of 10 rounds, AES-192 consists of 12 rounds, and AES-256 consists of 14 rounds. The iterative process of such steps is to improve the integrity of encryption.

Add round key

Add round key is the step which is followed through all rounds of encryption and decryption process, where the input and key generated from key generation algorithm are combined together.

For that the algorithm performs bitwise XOR operation on input and key.

Sub bytes

Input bytes are illustrated in the form of 4x4 matrixes. The input matrix also called state matrix. In sub bytes the algorithm holds a predefined matrix. This is stored in terms of lookup table. The look up table is named as S-BOX (Substitution Box). In the above mentioned step the each byte in input matrix is replaced by the each byte from the lookup table. This is shown in figure.2.

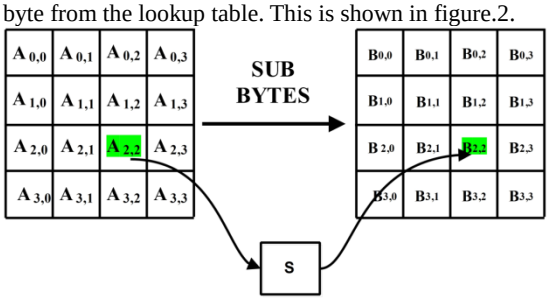


Figure 2: Process Diagram performing substitution bytes.

Shift rows

In this step each byte in the state matrix is shifted in a certain method, the shifting method is explained in figure.3.

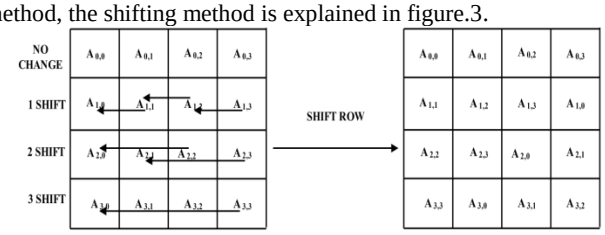


Figure 3: Process Diagram performing shifting of rows.

In the figure 2 it can be observed that the first row remains the same during the entire process. The second row undergoes complete byte shift by one sub block. Similarly the third and fourth row undergoes twice and thrice shifts respectively.

Mix column

During mix column operation each column from input state matrix is multiplied with the predefined constant matrix. The resulting column is illustrated as the corresponding column. So each input matrix will affect each column of resulting matrix.

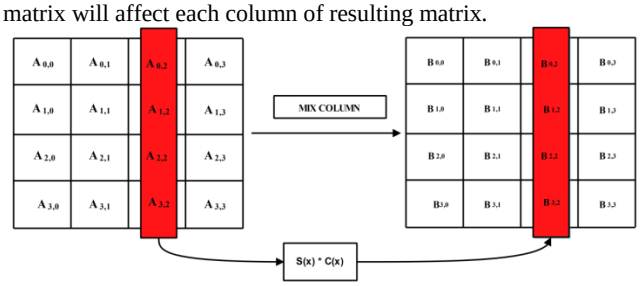


Figure 4: Process diagram performing mix columns

C) Secure Hash Algorithm (SHA)

After AES playing an important role in encryption for banking and medical data transmission, the objective towards encryption has increased for high security data transmission. So NIST researchers have approached for new standards for the same. This lead to implement the Secure Hash Algorithm standards. The SHA-1 algorithm is the revised and updated version of SHA-0 algorithm. It uses the input as 264 bit and generated output digest size is 160 bits. The produced digests are used in many security purposes. The noticeable difference between SHA-0 and SHA-1 is number rounds of mathematical calculation. The mathematical calculations performed in each round are shown Figure.5. To improve the confidentiality of SHA-1 algorithm the following calculations are performed such as swap, shift and addition. The successive additions performed in this algorithm ads additional complexity to the intruders. The message inputted into SHA-1algorithm is hashed by iterating the round function for 80 times. Ft is logical function and each Ft, $0 < t < 79$, operates on three 32-bit words and produces a 32-bit word as output. It is defined as follows.

$$F_t(B,C,D) = (B \wedge C) \vee (\sim B \wedge C) \quad (7)$$

$$F_t(B,C,D) = B \text{ xor } C \text{ xor } D \quad (8)$$

$$F_t(B,C,D) = (B \wedge C) \vee (B \wedge C) \vee (C \wedge D) \quad (9)$$

$$F_t(B,C,D) = B \text{ xor } C \text{ xor } D \quad (10)$$

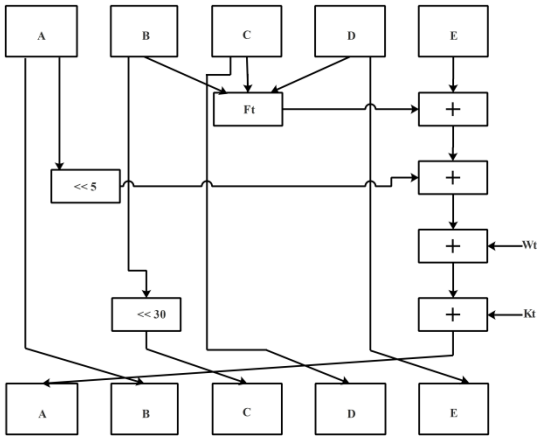


Figure 5: Functional block diagram of SHA1

SHA-2 algorithm has different groups such as SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256. SHA-2 algorithms are used to generate the message digests, the output digests are fixed in its length. To generate the fixed length digests; at the beginning of the algorithm it performs padding. Once the padding is completed much mathematical calculations are performed to generate the output digests. The number of rounds are depends on input size. The complexities of intrusion will increases based on the number of rounds. To improve that the revised version of SHA-2 algorithm consist more rounds. The number of rounds are depends the input bit size. The mathematical calculations performed in each round are shown in Figure.6.

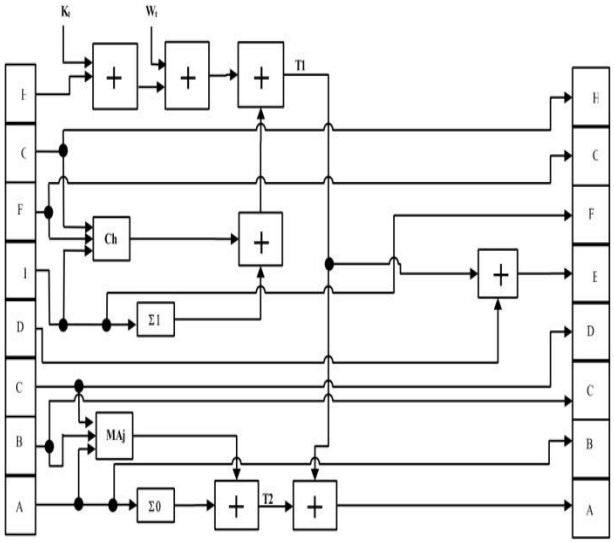


Figure 6: Functional block diagram of SHA2

Figure.7 represents the process involved in SHA-3 algorithm. The algorithm comprises of five steps namely: Theta, Rho, Pi, Chi and Iota. Each function performs different process on the data. The Theta function performs simple XOR and bitwise cyclic shift operations as shown in equation.11-13. The equations represent XOR operation (equation 11), then left circular shift on the five output lanes followed by right circular shift. This alternate left and

right shift between the lanes changes the bit positions (equation 12). Finally the XOR operation is performed between input and output lanes (equation 13). Rho and pi (equation 14) performs circular rotation on input data and divide those values by decimal value five. The resulting data given as input for chi step (equation 15) and iota step (equation 16). In iota step it performs bitwise XOR and AND operation alone.

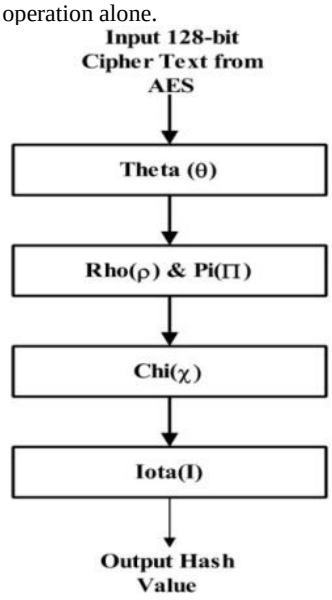


Figure 7: Block diagram of compressing unit in SHA3 algorithm

$$C[X] = A[X,0] \oplus A[X,1] \oplus A[X,2] \oplus A[X,3] \oplus A[X,4]$$
$$0 \le X \le 4 \quad (11)$$

$$D[X] = C[X-1] \oplus ROT(C[X+1,1]$$
$$0 \le X \le 4 \quad (12)$$

$$A[X,Y] = A[X,Y] \oplus D[X]$$
$$0 \le X \le 4 \quad (13)$$

$$B[Y,2X+3Y] = ROT(A[X,Y], r[X,Y])$$
$$0 \le X \le 4 \quad (14)$$

$$A[X,Y] = B[X,Y]((NOTB[X+1,Y])ANDB[X+2,Y])$$
$$0 \le X \le 4 \quad (15)$$

$$A[0,0] = A[0,0] \oplus RC \quad 0 \le X \le 4 \quad (16)$$

4. Result and discussion

AES Algorithm

To enrich the usage of RSA, AES, SHA-1, SHA-2 and SHA-3 algorithm, these algorithms are implemented in the FPGA. The simulations are performed in QUARTUS II.

Each process in all algorithms are separately implemented in behavioral model. Finally all process is combined using structural model.

During the simulation various parameters are noted like power consumed in the system, time taken to execute the code and number of logic elements taken by the executed code. For the comparison process the all algorithms are implemented in different FPGA devices like Cyclone II, III and Stratix II, III. SHA-2 algorithm has different groups such as SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224 and SHA-512/256. Here the SHA-256 is implemented in VHDL code for the comparison process. SHA-2 algorithms are used to generate the message digests, the output digests are fixed in its length.

To generate the fixed length digests; at the beginning of the algorithm it performs padding.

Once the padding is completed much mathematical calculations are performed to generate the output digests. The details of mathematical calculations are discussed in the background methodology.

Table I shows the power analysis of all algorithms.

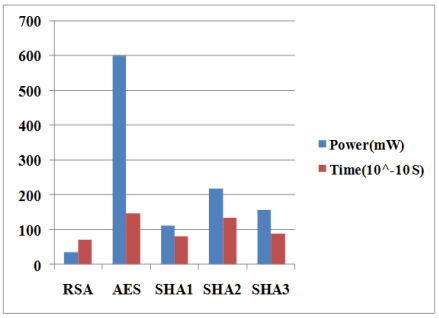
Table I: Power Analysis (mW)

	Cyclone II	Cyclone III	Stratix II	Stratix III
RSA	34.71	59.14	326.84	399.67
AES	601.80	318.43	769.27	421.35
SHA1	112.20	95.66	424.49	457.85
SHA2	218.18	151.34	489.32	456.02
SHA3	157.50	141.85	187.50	449.29

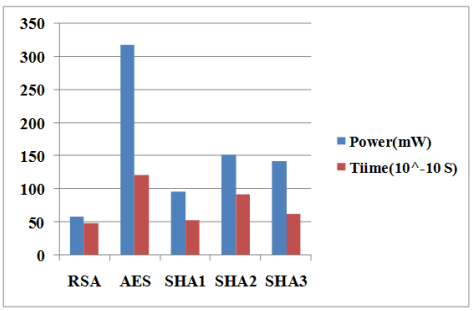
Table II: Time Analysis (nS)

	Cyclone II	Cyclone III	Stratix II	Stratix III
RSA	7.135	4.853	4.967	4.589
AES	14.605	12.147	11.209	9.709
SHA1	8.005	5.257	6.188	4.685
SHA2	13.498	9.267	9.231	7.094
SHA3	8.922	6.223	9.391	8.896

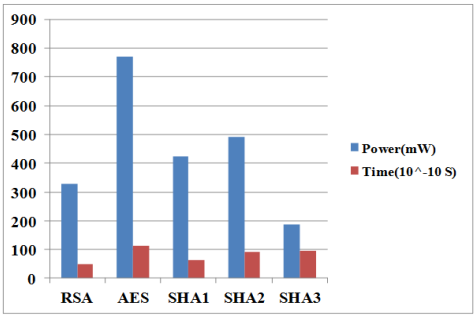
CYCLONE II



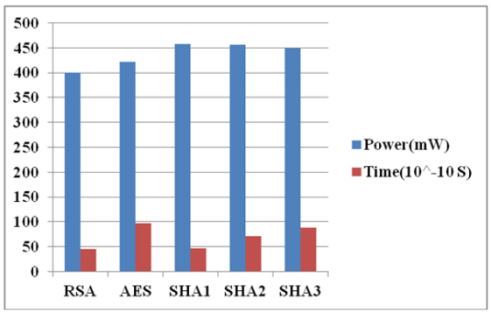
CYCLONE III



STRATIX II



STRATIX III



5. Conclusion

Many application areas like defense department and biomedical systems need secured data transmission. The application chosen for the work is the transmission of biomedical data. Several algorithms in the literature was reviewed and analyzed. The background methodology of the existing methods is presented in this paper. The encryption standard algorithms like RSA, DES, AES, SHA 1, SHA2 and SH3 are reviewed and implemented. The implementation was carried out in Quartus software using various FPGA kits. From the analysis and security level considered the AES and SHA3 algorithm dominates other algorithms. In future a new algorithm will be proposed.

References

[1] Amit T, Shriniwas D , Pritesh J & Akshay C, "Implementation of RSA Encryption Algorithm on FPGA", *American Journal of Engineering Research (AJER)*, Vol.4, No.6, (2015), pp.144-151.

[2] Landge AR & Ansari AH, "RSA algorithm realization on FPGA", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, Vol 2, No.7, (2013), pp.2323-2327.

[3] Muhammad II, Mamun BIR, Khandaker A & Sazzad H, "FPGA Implementation of RSA Encryption Engine with Flexible Key Size", *International Journal of Communications*, Vol.1, No.3, (2007), pp.107- 113.

[4] Sushanta KS & Manoranjan P, "FPGA Implementation of RSA Encryption System", *International Journal of Computer Applications*, Vol.19, No.9, (2011), pp.10-12.

[5] Velibor S, Branko D & Zeljko I, "Hardware Modules of the RSA Algorithm", *Serbian Journal of Electrical Engineering*, Vol.11, No.1, (2014), pp.121-131.

[6] Rohith SP & Mahesh C, "FPGA Implementation of 16 bit RSA Cryptosystem for Text Message", *International Journal of Computer Applications*, Vol.92, No.8, (2014), pp.1-5.

[7] Adriana NB, "Software Implementation of a Module for Encryption and Decryption Using the RSA Algorithm", *International Scientific Conference Electronics (ET)*, (2016), pp.1–4.

[8] Sangita AJ & Santosh SC, "A Modified RSA Algorithm to Enhance Security for Digital Signature", *International Conference and Workshop on Computing and Communication (IEMCON)*, (2015), pp.1-5.

[9] Ji Y & Hongbo K, "FPGA Implementation of Dynamic Key Management for DES Encryption Algorithm", *International Conference on Electronic & Mechanical Engineering and Information Technology*, (2011), pp.4795-4798.

[10] Wuling R & Zhiqian M, "A Hybrid Encryption Algorithm Based on DES and RSA in Bluetooth Communication", *Second International Conference on Modeling, Simulation and Visualization Methods*, (2010), pp.221-225.

[11] Jing W, Guo PJ & Hua Y, "Improved DES Algorithm based on Irrational Numbers", *IEEE Int. Conference Neural Networks & Signal Processing*, (2008), pp.632–635.

[12] Pradnya K & Phade GM, "Application of AES Algorithm for Data Security in Serial Communication", *International Conference on Inventive Computation Technologies (ICICT)*, (2016), pp.1–5.

[13] Guang LG, Quan Q & Rui Z, "Different Implementations of AES Cryptographic Algorithm", *IEEE 17th International Conference on High Performance Computing and Communications*, (2015), pp.1848–1853.

[14] Dilna V & Babu C, "Area Optimized and High Throughput AES Algorithm based on Permutation Data Scramble Approach", *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, (2016), pp.3056–3060.

[15] Savitha S & Yamuna S, "Implementation of AES Algorithm to Overt Fake Keys against Counter Attacks", *International Conference on Computer Communication and Informatics (ICCCI)*, (2016), pp.1-5.

[16] Pritamkumar NK & Vrushali GR, "Implementation of AES Algorithm on FPGA for Low Area Consumption", *International Conference on Pervasive Computing (ICPC)*, (2015), pp.1-4.

[17] Senthil Kumar J & Mahalakshmi C, "Implementation of Pipelined Hardware Architecture for AES Algorithm using FPGA", *International Conference on Communication and Network Technologies (ICCNT)*, (2014), pp.260–264.

[18] Nacci AA, Rana VSD & Santambrogio MD, "An open-source, efficient and parameterizable hardware implementation of the AES algorithm", *IEEE International Symposium on Parallel and Distributed Processing with Applications*, (2014), pp.85-92.

[19] Mazen EM, Salma H & Mohamed AAEG, "Real-Time Efficient FPGA Implementation of AES Algorithm", *IEEE International SOC Conference*, (2013), pp.203–208.

[20] Luminita S and Petre DM, "On the Substitution Method of the AES Algorithm", *International Symposium on Signals, Circuits and Systems (ISSCS)*, (2013), pp.1–4.

[21] Kshirsagar RV & Vyawahare MV, "FPGA Implementation of High speed VLSI Architectures for AES Algorithm", *Fifth International Conference on Emerging Trends in Engineering and Technology*, (2012), pp.239–242.

[22] Junfeng C & Mohammed B, "Low Area Memory-Free FPGA Implementation of the AES Algorithm", *22nd International Conference on Field Programmable Logic and Applications (FPL)*, (2012), pp.623–626.

[23] Vanitha M & Sakthivel R, "Highly Secured High Throughput VLSI Architecture for AES Algorithm", *International Conference on Devices, Circuits and Systems (ICDCS)*, (2012), pp.403–407.

[24] Atul MB, Kshirsagar RV & Vyawahare MV, "FPGA Implementation of AES Algorithm", *3rd International Conference on Electronics Computer Technology*, Vol.3, No.1, (2011), pp.401–405.

[25] Magnus S & Ricardo C, "Efficient FPGA Implementation of the SHA-3 Hash Function", *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, (2017), pp.86–91.

[26] Muhammad A, Muhammad A, Nasir M & Arshad A, "Compact Hardware Implementation of SHA-3 Finalist Blake on FPGA", *IEEE 9th International Conference on Emerging Technologies (ICET)*, (2013), pp.1–5.

[27] Fatma K, Belgacem B, Mohsen M & Rached T, "An FPGA implementation of the SHA-3: The BLAKE Hash Function", *International Multi-Conferences on Systems, Signals & Devices (SSD13)*, (2013), pp.1–5.

[28] Kuntoro A, Arimaz H & Made S, "SHA-2 and SHA-3 Based Sequence Randomization Algorithm", *2nd International Conference on Science and Technology-Computer (ICST)*, (2016), pp.150–154.

[29] Kazuyuki K, Jun I, Miroslav K & Eric XG, "Prototyping Platform for Performance Evaluation of SHA-3 Candidates", *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, (2010), pp.60–63.

[30] Jens-Peter K, Panasayya Y, Kishore KS, Bilal H, Susheel V & Smriti G, "Lightweight Implementations of SHA-3 Finalists on FPGAs", *International Conference on Cryptology*, (2011), pp.270-289.

[31] Bernhard J & Jurgen A, "Area-efficient FPGA Implementations of the SHA-3 Finalists", *International Conference on Reconfigurable Computing and FPGAs*, (2011), pp.235–241.

[32] Yusuke A, Naoki F & Shuichi I, "Design Trade-offs in SHA-3 multi-message hashing on FPGAs", *TENCON 2014-2014 IEEE Region 10 Conference*, (2014), pp.1-5.

[33] Bernhard J & Marc S, "Hobbit - Smaller But Faster Than A Dwarf: Revisiting Lightweight SHA-3 FPGA Implementations", *International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, (2016), pp.1-7.