IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 12, Issue 1, Ver. II (Jan.-Feb. 2017), PP 54-59
www.iosrjournals.org

A Survey on Various Lightweight Cryptographic Algorithms on FPGA

Chaitra B[1], Kiran Kumar V.G[1], Shantharama Rai C[2]
[1](Electronics & Communication Engineering, Sahyadri College of Engineering& Management, India)
[1](Associate Professor Electronics & Communication Engineering, Sahyadri College of Engineering & Management, India)
[2](Principal,, AJ Institute of Engineering &Technology, Mangaluru, India)

Abstract : *In today's rapid growing technology, digital data are exchanged very frequently in seamless wireless networks. Some of the real time applications examples which are transmitted quickly are voice, video, images and text but not limited to high sensitive information like transaction of creditcard, banking and confidential security numbers/data. Thus protection of confidential data is required with high security to avoid unauthorised access to Wireless networks. This can be done by a technique called 'Cryptograhy' and there are two crytography techniques available (such as symmetrical & asymmetrical techniques). The focus in this paper would be on Lightweight symmetric crytography. Lightweight cryptography is used for resource-limited devices such as radio frequency identification (RFID) tags, contactless smart cards and wireless sensor network. In this paper comparative study of selected lightweight symmetric block ciphers such as AES, PRESENT, TEA and HUMMINGBIRD is presented.*

Keywords: *LFSR, Keys, Encryption*

Introduction

Cryptography is an art of security technique where messages are encoded in a non-readable form. In simple words, it is nothing but a technique used in the protection of data during the transmission from sender to receiver and unauthorized access is denied. Therefore, security and confidentiality is very much required in this aspect. The focus and discussion in this paper would be on various techniques of "Lightweight Cryptography (LWC)" and analyzing which would be the best methodology. Lightweight Cryptography is one of the emerging research areas in cryptography. This covers cryptographic algorithms intended for use in devices with low or extremely low resources. Thus, it is used in RFID tags, contactless smart cards, sensors etc. Lightweight cryptography does not determine strict criteria for classifying a cryptographic algorithm as lightweight, but the common features of lightweight algorithms are extremely low requirements to essential resources of target devices. Considering area and energy consumption are the important measures to evaluate the LWC properties. Also, we highlight some constraints and recommendations of lightweight algorithms.
We will discuss some of the methodology of various Lightweight cryptographic algorithms below.

I. AES

AES is a better choice for many block ciphers. It can be used with the higher level of security throughput with less area. AES was developed by two scientists Joan and Vincent Rijmen in 2000. It is a symmetric block cipher with a block size of 128 bits and Key lengths can be 128 bits, 192 bits, or 256 bits called AES-128, AES-192, and AES-256, respectively. AES-128 uses 10 rounds, AES-192 uses 12 rounds, and AES-256 uses 14 rounds. Fig.1 shows block diagram of AES block cipher. AES is a symmetric block cipher it uses four steps of operation. The last round consists of only 3 steps.

- Sub Bytes
- Shift Rows
- Mix Columns
- AddRoundKey

1.1 Sub byte: It is the byte substitution on the input data; each byte in the array is updated using an 8-bit substitution box, called as Rijndael S-box. The S-box used is derived from the multiplicative inverse over gf (2^8), known to have good non-linearity properties. The affine transformation is used to generate the Sub Box, which is the lookup table from which the values are substituted. The procedure to be followed to substitute the bytes in the matrix is:

- Select any element says i, j which is in hexadecimal notation.
- The S-box element in the i th row and j th column is to be selected and substituted in its place.

1.2 Shift rows: It involves rotating the rows of the input matrix circularly upwards.

1.3 Mix column: Modulo Matrix Multiplication of the input matrix with the data matrix.
1.4 Add round key: This is a transformation bitwise XOR operation that combines the current state data block and the round key.

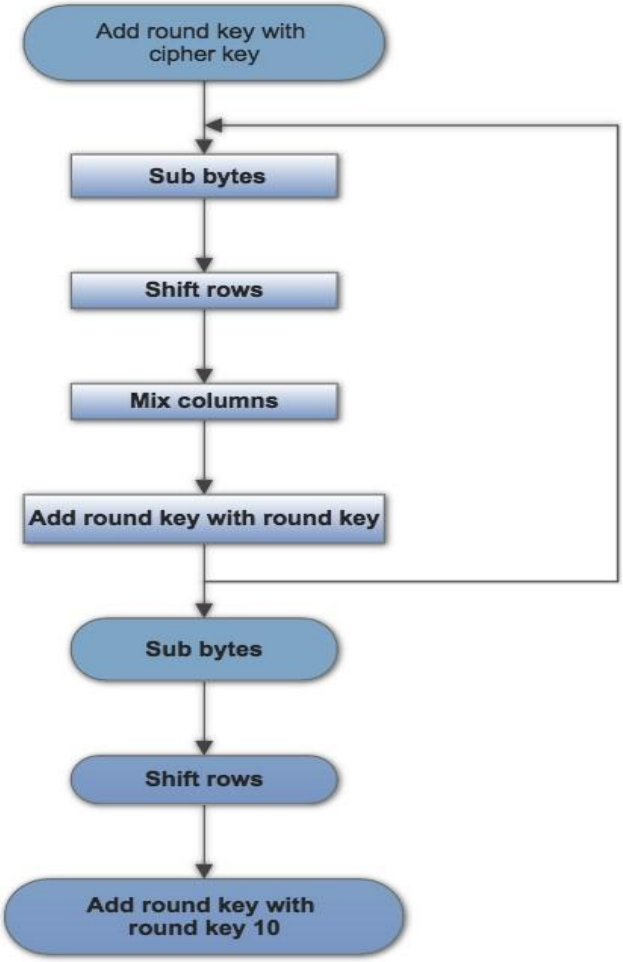

Fig 1. AES Implementation

Key Generation Algorithm:
Each round key array is generated as the product of previous round key. In the sequence of 11 keys we need a matrix of 4 rows and 44 columns.

1. Generating the columns C4m: On the hardware scale, it can be implemented by shift registers for Rotation Word Transformation whereas the Substitution bytes' transformation will be implemented by a ROM module with the S-box as look up table and XOR gates for the addition

   W (i) =W (i-4) XORW (i-1) *I* is not a multiple of 4

2. Generating the Columns C4m+1; 4m+2; 4m+3 this on the hardware scale can be implemented using XOR gates alone and registers to store the values.

   W (i) =W (i-4) XORT (W (I-1)) *I* is a multiple of 4.

II. PRESENT

Every design of the lightweight cryptography must suit better security, efficiency and minimum cost. PRESENT is such a hardware-computed ultra-lightweight block cipher designed to achieve less area and power constraints.Fig.2 shows the block digram of PRESENT encryption algorithm. It is best suitable SPN (substitution permutation network) structure. It has a 64-bit block size 80 or 128-bit key size and performs 31 rounds. Each round consists 3 steps they are Add round key, Substitution, and permutation.

2.1. In the first step XOR operation of plain text and key takes place.

2.2. The result is then Substituted Using S-box. The substitution layer consists of 16 S-Boxes that each has 4-bit input and 4-bit output (4x4). The S-Box is given in hexadecimal notation shown in the Table 1.

2.3. The outcome is applied to Permutation Using P-layer. Bit *i*of STATE is moved to bit position P (*i*).

The P box is shown in the Table 2.

Table 1: S box layer of PRESENT

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S[x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table 2: Permutation layer of PRESENT

i	P(i)	i	P(i)	i	P(i)	i	P(i)	i	P(i)	i	P(i)	i	P(i)	i	P(i)
0	0	8	2	16	4	24	6	32	8	40	10	48	12	56	14
1	16	9	18	17	20	25	22	33	24	41	26	49	28	57	30
2	32	10	34	18	36	26	38	34	40	42	42	50	44	58	46
3	48	11	50	19	52	27	54	35	56	43	58	51	60	59	62
4	1	12	3	20	5	28	7	36	9	44	11	52	13	60	15
5	17	13	19	21	21	29	23	37	25	45	27	53	29	61	31
6	33	14	35	22	37	30	39	38	41	46	43	54	45	62	49
7	49	15	51	23	53	31	55	39	57	47	59	55	61	63	63

Key schedule:

PRESENT can take keys of either 80 or 128 bits they are referred as PRESENT-80 and PRESENT-128. In this paper we focuson 80-bit keys are represented as k79, k78 . . . k0 and they are stored in key register K. At round i the 64-bit round key Ki = k63k62 . . . k0 consists of the 64 leftmost bits of the current contents of register K. Thus at round i we have Ki = k63k62 . . . k0 = k79k78 . . . k16 will be extracted then key register K = k79k78 . . . k0 is updated as

1. Key register is rotated by 61 positions to the left. [K79k78 . . . k1k0] = [k18k17 . . . k20k19]

2. The left most four bits are passed through the PRESENT S-box [k79k78k77k76] = S [k79k78k77k76]

3. The round counter value is exclusive ORed with bits of K with least significant bit of round counter on the right**.**

$$\textbf{[k19k18k17k16k15] = [k19k18k17k16k15]} \oplus \textbf{roundcounter}.$$

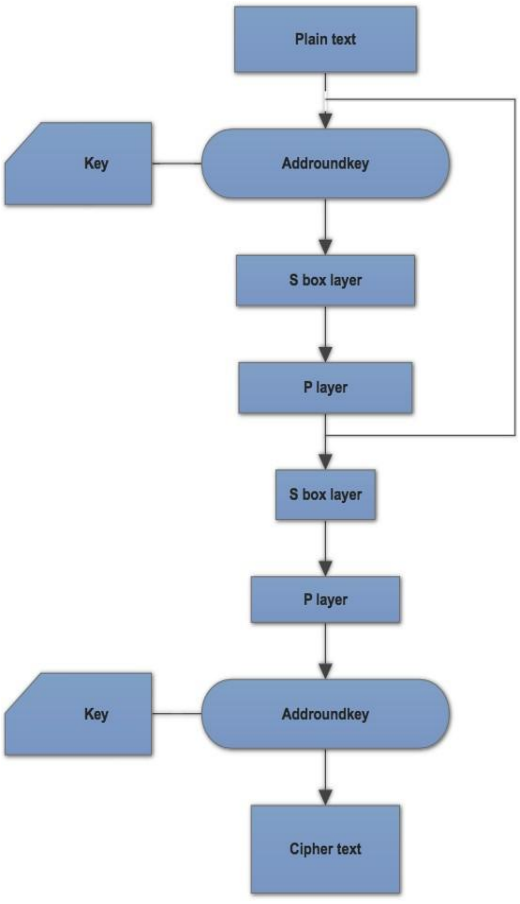

Fig 2. Block diagram of PRESENT cipher

III. TEA

Both security and simplicity of implementation is equally important and hence TEA is an excellent choice in cryptography. The Tiny Encryption Algorithm is block cipher and it is simpler as a few lines of the code. It is fast, secure and simple in description and implementation than IDEA even though it uses same algebraic mixed group's technique. As the confidentiality of the data is more important TEA is secure and needs minimal storage space. Tea is highly resistant to differential cryptanalysis and achieves complete diffusion. The cipher was developed by Wheeler and Needham in 1994.Fig.3 shows the block diagram of TEA.

The TEA takes 64 bit (block size) data bits using 128 bit keys with 32 rounds. This cipher starts with a 64 bit data block that is split up into two 32 bit blocks in which the block on the left side is called as L and the block on the right side is called as R. These blocks are swapped per Round.

Key schedule:

128 bit key is split into four 32 bit sub key Ki where i= 0 to 3 and it uses delta, delta is defined as a constant, 2^{32}/ (golden ratio), which is 2654435769 as an integer. Multiples of delta are used in each round (mod 2^32).All addition operations are mod 2^{32}.

3.1. The one half P1 [i-1] say R goes through a left shift of 4 and then is added to K [0]

3.2. R is added to Delta

3.3. R goes through a right shift of 5 and then is added to K [1]

An XOR operation is then applied to the result of those three operations and finally, the result of
The XOR operation is added to L (P0 [i-1]).it produces the half of the block cipher for the next round. Similar operations are performed for the next half round P0 [i-1] function.

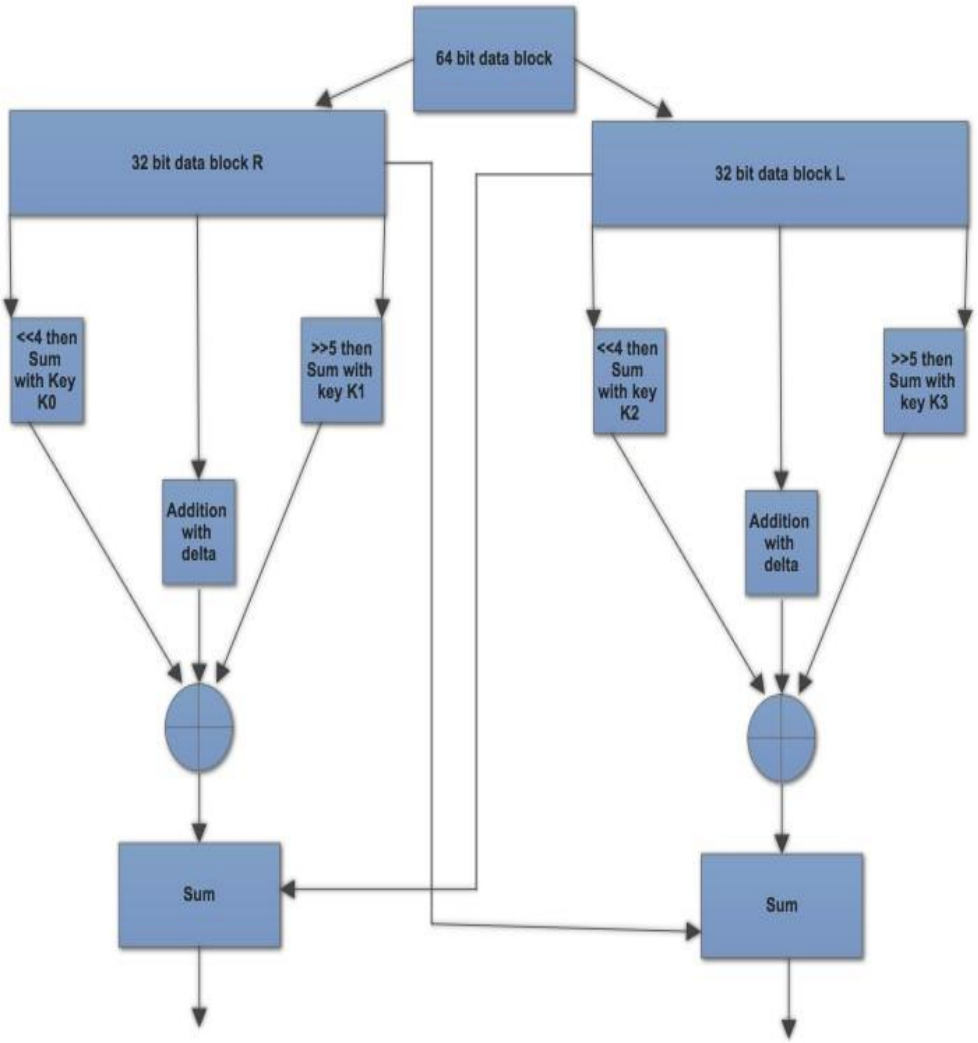

Fig 3. Implementation of TEA encryption algorithm

IV. Hummingbird

It is important to realize that ultra-lightweight cryptographic algorithm Hummingbird shown in fig.4, is designed for resource-constrained devices. The design of the Hummingbird cryptographic algorithm is motivated by the well- known Enigma machine taking into consideration of both security and efficiency. It is the combined block cipher and stream cipher. It has 16 bit block size, 256 bit key size and 80 bit internal state.

The 16-bit plain text is PTi and the first internal state register RS1.First step in the encryption is mod 2^{16} addition of plain text and the content of internal state register. It is encrypted by the first block cipher EK1.That is achieved by Ex-OR operation of added result with key followed by the substitution and permutation. The procedure of encryption is repeated 3 times and the output EK4 is corresponding cipher text CTi.

The states of the four internal state registers will also be updated in an unpredictable way based on their current states, the outputs of the first three blocks and the state of the LFSR.

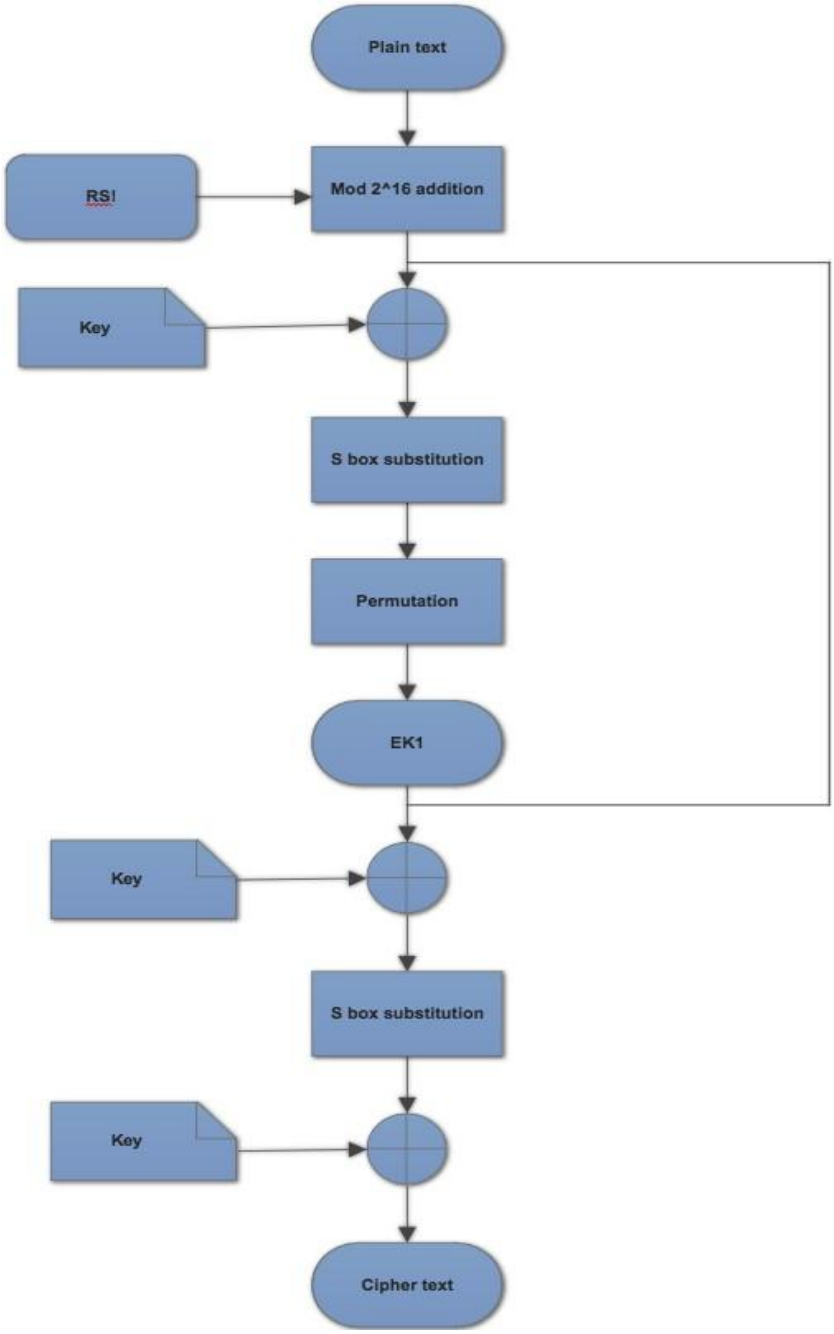

Fig 4. The Hummingbird encryption process.

V. Results and Analysis:

Table 3 shows the comparative analysis of the various lightweight ciphers and fig.5 shows the comparison graph of the various ciphers. It is seen that TEA has a better throughput while PRESENT-80 requires less number of slices and TEA has a better efficiency.

Table 3. Comparison between Different Types of Lightweight Cryptographic Algorithms

Cipher	Key Size	FPGA Device	Throughput	Slices(GEs)	Efficiency
PRESENT - 80	**80**	**Spartan3XC35400**	**516**	**176**	**2.93**
PRESENT - 128	**128**	**Spartan3XC35400**	**508**	**202**	**2.51**
TEA	**128**	**Spartan3XC35200**	**19.52**	**1140**	**0.017**
AES	**256**	**Spartan-II XC2S30-6**	**166**	**522**	**0.32**
HUMMINGBIRD	**256**	**Spartan3XC3S200-5**	**160.4**	**273**	**0.59**

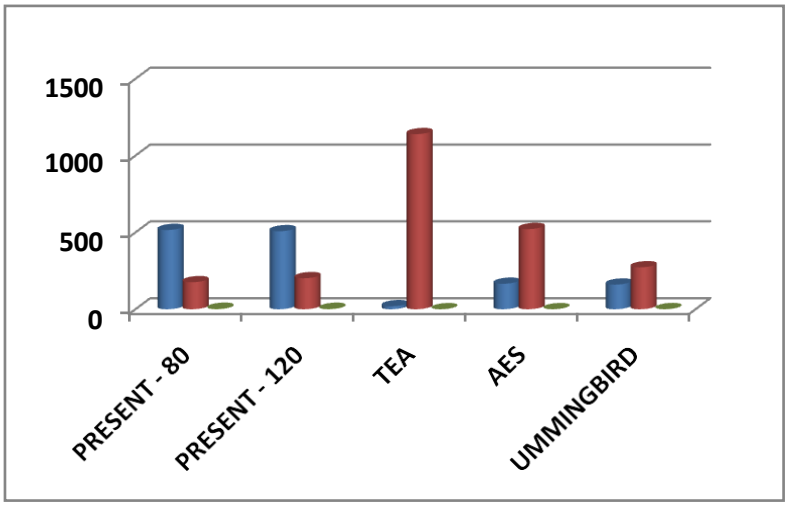

Fig 5. Comparison graph of various ciphers.

VI. Conclusion

In today's wireless network world LWC plays an important role with resource constraint devices in providing security. There are different methodologies of LWC where each methodology has its own properties and compared in the table. Analyzing the comparison of algorithms- AES, PRESENT, TEA, HUMMINGBIRD, it is shows that PRESENT has higher efficiency, less area and power consumption, thereby reducing cost. This algorithm also provides adequate security and also to mention that PRESENT uses 31 rounds for 256 block size.It is more applicable for as cryptographic algorithm for resource constrained devices.

Reference

[1]. "VLSI design of secure cryptographic algorithm", Revini s. shende, Mrs. Anagha y. Deshpande, International Journal of Engineering research & applications (IJERA) Vol. 3 March-April 2013.
[2]. "A Survey on Various Cryptography Analysis"by Mitali, Vijay Kumar and Arvind Sharma, Volume 3, Issue 4, July-August 2014.
[3]. "Encryption Using Different Techniques" by Dimple1 - Vol. 2, No.1, January-February-2013.
[4]. "Image Encryption using Different Techniques for High Security Transmission over a Network", by Mohammad Sajid, Qamruddin Khizrai, Prof. S.T. Bodkhe - Volume 2, Issue 4, June-July, 2014.
[5]. Amrutha k, Jayachandra Naidu V., "Advanced Encryption Standard Algorithm Implementation Using Verilog HDL" Vol. 04, Article 06117; July 2013.
[6]. Sergey panasenko, Sergey smagin by" Lightweight cryptography: Underlying Principles & Approaches", International Journal of Computer Theory & Engineering, Vol. 3, No. 4 August 2011.
[7]. Kiran Kumar V.G., Sudesh Jeevan Mascarenhas, Sanath Kumar, Rakesh J Paris, "Design and Implementation of Tiny Encryption Algorithm" Vol. 5, Issue 2, June 2015.
[8]. A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher".
[9]. Mohamad Sbeiti, Michael Silbermann, Axel Poschmann, Christof Paar., "Design Space Exploration of PRESENT Implementation For FPGA".
[10]. Thomas Eisenbarth, Christof Paar and Axel Poschmann, Sandeep Kumar, Leif Uhsadel "A Survey of Lightweight-Cryptography Implementations".
[11]. "Design and simulation of AES algorithm- Encryption using VHDL" Mital Maheta, 2014 IJEDR Volume 2, Issue1, 2014.
[12]. "Study of Hummingbird Cryptographic Algorithms based on FPGA implementation", by Reena Bhatiya, IJCSIT, International Journal of computer Science & Information Technologies, Vol. 5(3),2014.