IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 10, Issue 3, Ver. II (May - Jun.2015), PP 18-24
www.iosrjournals.org

Design and Implementation of Hybrid Cryptosystem using AES and Hash Function

Vanishreeprasad. S[1] Mrs. K N Pushpalatha[2]
[1](MTECH student, Dept. Electronics & Communication, Dayananda Sagar College of Engineering, India)
[2](Associate professor, Dept. Electronics & Communication, Dayananda Sagar College of Engineering, India)

Abstract : *Secure data communication is of a key concern in today's rapidly growing world. Various security mechanisms are developed in order to achieve the data security. Cryptography is one among them. It is the study of mathematical techniques that are related to the aspects of information security such as confidentiality, data integrity, authentication, and availability. The proposed architecture integrates the cryptographic algorithms, Advanced Encryption Standard algorithm (Symmetric) and the Hash function, SHA-2 to improve the data security to a greater extent. The design is synthesized using Xilinx ISE software and implemented on Virtex-5 xc5vlx110t-2ff1136 board.*

Keywords : *Advanced Encryption Standard (AES), Cryptography, Cryptographic Algorithms, Hash functions; Secure Hash Algorithm (SHA-2).*

I. INTRODUCTION

In today's fast moving world, transmission of electronic data safely from one point to another is of a key concern. Security becomes an important aspect while transmitting the data. Though there exists many algorithms to carry on the secure data communication, security is still a critical aspect to achieve. Crypto-system is the one that can help us to deal with the security related problems by encrypting the data in sender and decrypting it in the receiver [1]. Cryptography involves three distinct mechanisms:

Symmetric-key encipherment, Asymmetric-key encipherment and Hashing. Symmetric-key encipherment uses a single secret-key for both encryption and decryption. DES, AES are some of the symmetric crypto algorithms.

Asymmetric-key encipherment uses two keys instead of one: one public key and one private key. Rivest-Shmir–Adleman (RSA) and Elliptic curve cryptography (ECC) are two representatives of asymmetric crypto system. Cryptographic Hash function is a mathematical transformation that takes a message of arbitrary length and computes a fixed-length (short) number out of it. MD-5, SHA (SHA-0, 1, 2, 3) are some of the hash functions [2]. AES was published by National Institute of Standards and Technology (NIST) in 2001. Later Rijndael algorithm was selected as AES algorithm [3]. The SHA-2standard [4] supersedes the existing SHA-1, for computing a condensed representation (message digest) of electronic data.

In this paper, an architecture that integrates AES algorithm and SHA-2 hash algorithm is presented. The proposed design provides high security in terms of complexity.

The paper is organized as follows: Section II elucidates about the work related to the proposed system. Section III describes the AES algorithm used in the project. Section IV discusses the Key Expansion process for AES. Section V describes SHA-2 Hash algorithm. Section VI presents the proposed architecture for the integration of AES with SHA-2. Results and Discussions are given in section VII. Finally, in Section VIII brief conclusion is drawn.

II. RELATED WORK

Abhijith.P.S and Mallika srivastava [5] in their paper presented the hardware implementation of AES algorithm using Xilinx– virtex5 FPGA. In [6], a hybrid encryption method that combines both symmetric and asymmetric cryptographic algorithms to provide high security with minimized key maintenance is proposed. M.Meenakumari, G.Athisha [7] in their paper, have combined the Encryption algorithm of AES and the MD-5 hash function in order to realize the data integrity and confidentiality.

Previously many modified [8, 9] and much efficient hardware architectures [10] are developed for AES implementation. Adnan Abdul-Aziz Gutub, Farhan Abdul-Aziz Khan [11], in their paper has proposed a hybrid crypto system that uses the benefits of both symmetric key and public key cryptographic methods. Also a single method [12] that will ensure the Confidentiality, Integrity, Availability and Authentication of the message to be transmitted was introduced by Neeta Wadhwa, Syed Zeeshan Hussain and S.A.M Rizvi.

III. AES (ADVANCED ENCRYPTION STANDARD)

The Advanced Encryption Standard (AES) is a FIPS-approved cryptographic algorithm that can be used to protect the electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) the data or information. Encryption process converts the data into an unintelligible form called ciphertext. Decryption converts back the ciphertext into its original form, called plaintext. The AES algorithm uses cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt the data in blocks of 128 bits. Encryption and decryption processes of AES are explained separately as follows:

3.1. Encryption process

The Encryption process of Advanced Encryption Standard algorithm for the proposed design is presented in Fig. 6. It consists of a number of transformations that will be explained later and these transformations are applied consecutively over the data block bits, in a fixed number of iterations, called rounds. The number of rounds depends on the length of the key used for the encryption process. Here user use 128-bit key so the number of rounds are 10. If the key length is 192-bit, then the number of rounds will be 12.

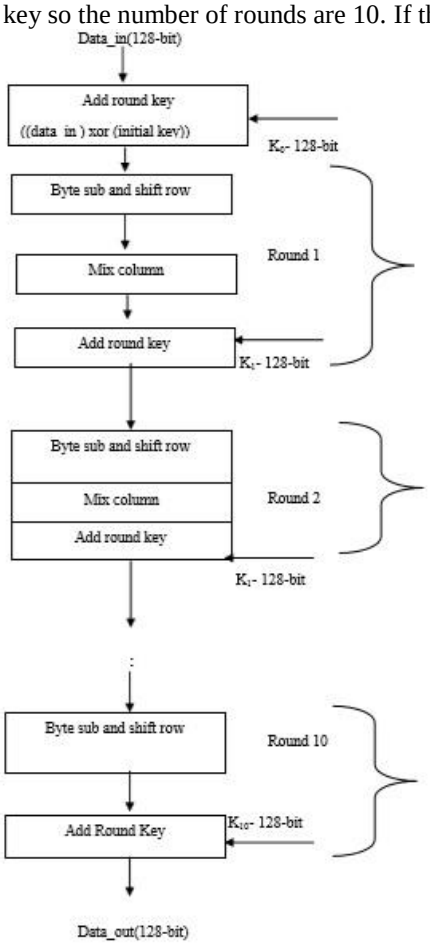

Fig.1 Encryption Process

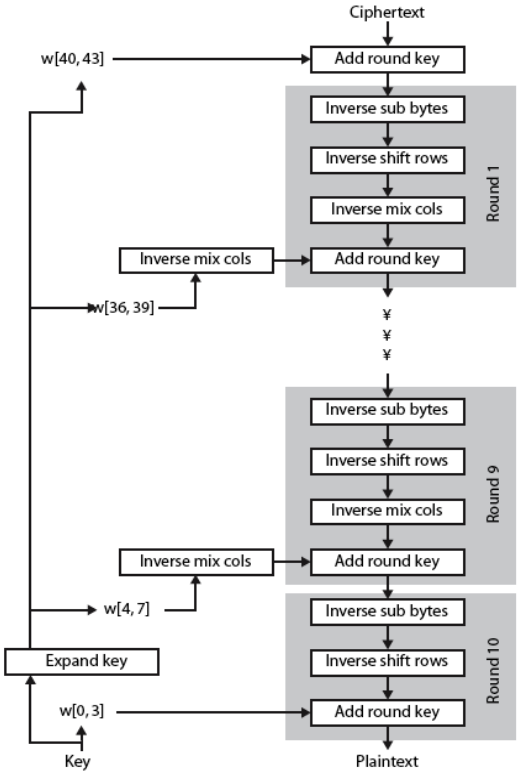

Fig. 2 Decryption process

3.2. Decryption process

The Decryption process of Advanced Encryption Standard algorithm is presented in Fig. 2. This is a process which is the direct inverse of the Encryption process. All the transformations applied in Encryption process are inversely applied to this process. Hence the last round values of both the data and key in encryption are the first round inputs for the Decryption process and this goes on in the decreasing order.

The operations or transformations of AES algorithm for encryption and decryption can be explained as follows:

3.3. Sub Byte and Inverse Sub Byte Transformation

In the Sub Bytes step, each byte in the state matrix is replaced with a Sub Byte using an 8-bit data from the Rijndael S-Box. In the Inverse Sub Bytes step, each byte in the cipher matrix is replaced with corresponding Inverse Sub Byte. Sub Byte operation will provide the non-linearity in the cipher. The S- Box used is derived from the multiplicative inverse over Galois Field (2^8) [13]. Fig. 3 shows sub-bytes operation.

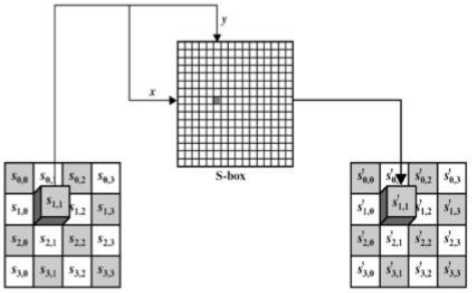

Fig. 3 Sub-bytes Transformation

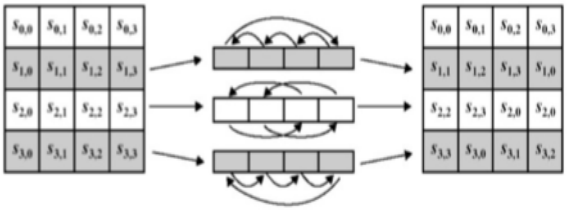

Fig.4 (a) Shift Rows Transformation

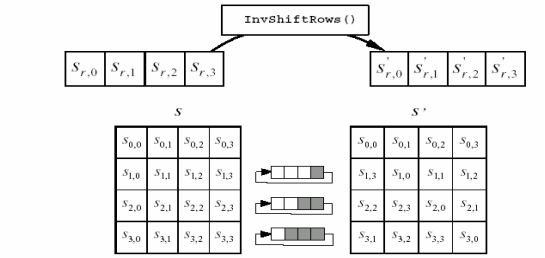

Fig.4 (b) Inverse Shift Rows Transformation

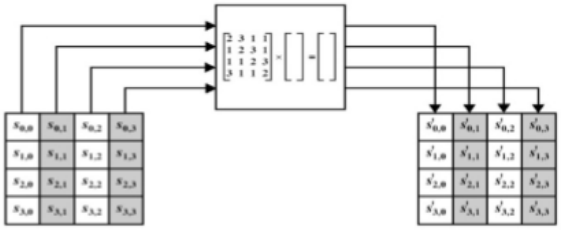

Fig. 5 Mix Columns Transformation

3.4. Shift Row and Inverse Shift Row Transformation

The Shift Rows transformation will perform the cyclic shifts of the bytes in each row by certain offset to the left. For AES, the first row remains unchanged. Each byte of the second row is shifted by one to the left. Similarly, the third and fourth rows are shifted by two and three respectively. Inverse Shift Row transformation does the same shift operation towards right. Fig.4 (a) and (b) shows the Shift Row and inverse shift row operation respectively.

3.5. Mix Column and Inverse Mix Column Transformation

This operation is basically a substitution but it makes use of arithmetic of GF (2^8). Each column is operated on individually. Here each byte of a column is mapped into a new value that will be a function of all four bytes in the column. Each element of the product matrix is the sum of products of elements of one row and one column. Here the individual additions and multiplications are performed in Galois Field (2^8). The inverse mix columns are performed in similar way but with different values in the matrix. Fig. 5 shows the operation of mix columns.

3.6. Add Round Key Transformation

In this operation, bitwise exclusive-or (XOR) operation is performed between outputs from Mix Column and Round Key. For AES-128, 128 bit XOR operations are performed.

IV. AES KEY EXPANSION

The AES key expansion algorithm takes the input which is a 4-word key and produces a linear array of 44 words. Each round uses 4 of these words. Each word contains 32 bytes which means each sub-key is 128 bits long. The key is copied into the first four words of the expanded key. The remainder of the expanded key is filled in four words at a time. Each added word w[i] depends on the immediately preceding word, w[i−1], and the word four positions back w[i−4]. In three out of four cases, a simple XOR is used. For a word whose position in the w array is a multiple of 4, a more complex function is used. Fig. 5 illustrates the generation of the first eight words of the expanded key using the symbol g to represent that complex function. The function g consists of the following subfunctions:

1. RotWord performs a one-byte circular left shift on a word. This means that an input word [b0, b1 ,b2, b3] is transformed into [b1, b2 , b3, b0].
2. SubWord performs a byte substitution on each byte of its input word, using the s-box described earlier.
3. The result of steps 1 and 2 is XORed with round constant, Rcon[j].

The round constant is a word in which the three rightmost bytes are always 0. Thus the effect of an XOR of a word with Rcon is to only perform an XOR on the leftmost byte of the word. The round constant is different for each round and is defined as Rcon[j] = (RC[J], 0,0,0), with RC[1]= 1, RC[j]= 2•RC[j−1] and with multiplication defined over the field GF(2^8).

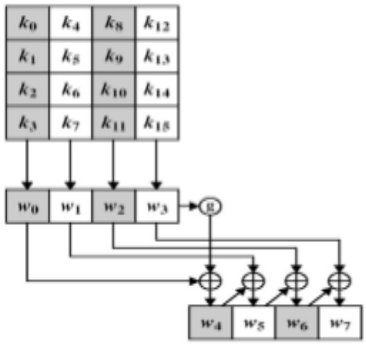

Fig. 6 Key Expansion

V. SHA-2 HASH ALGORITHM

SHA-2 is a type of cryptographic hash functions that is designed by the NSA (U.S. National Security Agency). SHA stands for Secure Hash Algorithm. Cryptographic hash functions are the mathematical operations that run on the digital data. A person can determine the data's integrity, by comparing the computed "hash" to a known and expected hash value. SHA-256 can accept messages with arbitrary lengths up to 264-bit. The Hash computation produces a final digest message of 256 bits that depends upon the input message, composed by multiple blocks of 512-bit each. This input block is expanded and it is fed to the 64 cycles of the SHA-256 function in words of 32-bit each.

5.1. Preprocessing

In SHA-256, the message to be hashed is first padded so that its final length becomes a multiple of 512-bit. The n-bit message is padded so that a single 1-bit is added into the end of the message. Then, 0 bits are added to make the length of the message congruent to 448 modulo 512. Then a 64-bit representation of n is appended to the result of the padding. Thus, the result message is a multiplicity of 512-bit. This message is denoted here as M(i). M(i) message blocks are passed individually to the message expander. Padding can be represented as shown in Fig.7.

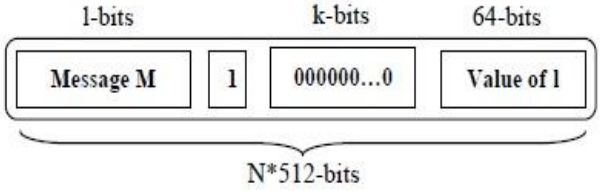

Fig. 7 Message preprocessing

5.2. Logical Functions

In SHA-256 algorithm, six logical functions are used that operates on 32-bit values:

$Ch(x, y, z) = (x \wedge y) \oplus (\sim x \wedge y)$ (1)

$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$ (2)

$\Sigma 0(x) = ROTR^{2}(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x)$ (3)

$\Sigma 1(x) = ROTR^{6}(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x)$ (4)

$\sigma 0(x) = ROTR^{7}(x) \oplus ROTR^{1}(x) \oplus SHR^{3}(x)$ (5)

$\sigma 1(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x)$ (6)

Where ^, ~ and $\oplus$ are the bitwise AND, NOT and XOR operations. ROTR and SHR are the rotate right and shift right functions respectively.

5.3. Hash Computation

The message, M is expended by a message Scheduler according to the following function:

For j = 0 to 15: $W = M_j^{(i)}$ and For j = 16 to 63{ $Wj = \sigma 1(Wj-2) + Wj-7 + \sigma 0(Wj-15) + Wj-16$ }

For i=1 to N, {Initialize registers a, b, c, d, e, f, g, h with the $(i-1)^{st}$ intermediate hash value}

Apply the following compression function to registers a-h:

For j= 0 to 63{ $T_1 = h+\Sigma_1(e)+Ch(a, b, c)+K_j+ W_j$

$T_2 = \Sigma_0(a) + Maj(a, b, c)$

h = g , g = f, f = e, e = d+T1
d = c, c = a, b = a, a = T1+T2}
i^{th} intermediate hash:
{ $H_1^{(i)} = a+H_1^{(i-1)},\ldots,H_8^{(i)} = h+H_8^{(i-1)}$ }
The hash of M: $H^{(N)} = (H_1^{(N)}, H_2^{(N)},\ldots, H_8^{(N)})$
Using the above logical functions and the equations, message digest is computed.

VI. PROPOSED ARCHITECTURE

6.1. Hybrid Cryptosystem

The proposed architecture i.e., Hybrid cryptosystem is represented in the following fig. It depicts the integration of AES algorithm with the SHA-2 hash function. The AES and SHA-2 algorithms are explained in previous sections. Accordingly the two algorithms are designed and integrated as shown in the Fig.8.

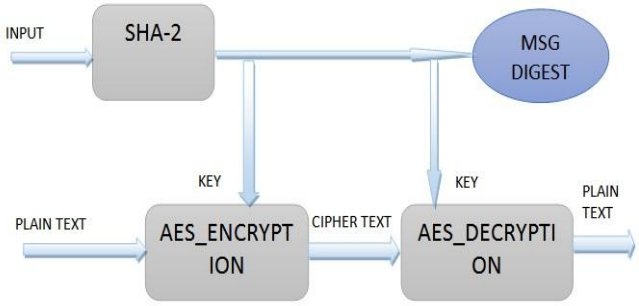

Fig. 8 Hybrid Cryptosystem

An input of arbitrary length is given to the SHA-2 module. A message digest of fixed length is generated which is 256 bits. This message digest is used in the encryption and decryption process as key. As shown in the fig.8, after generating the message digest, it is given as a key for the encryption of the plain text which in turn generates a cipher text. Later by making use of same key, decryption is performed to retain back the original plain text. AES itself is a strong security mechanism. Since SHA-2 is being used here along with the AES, this design ensures higher security since complexity of the design increases. Here security is given in terms of complexity.

VII. RESULTS AND DISCUSSIONS

This section discusses the results of the architecture designed. Hybrid Cryptosystem is synthesized on Xilinx FPGA target device using virtex xc5vlx110t-2ff1136 with -2 Grade speed. The device utilization summary is shown in TABLE. 1. As can been seen from the table, the proposed architecture utilizes 1911 slice registers which accounts for about 2% of the slice registers available and 5% of the slice LUTs available. TABLE.2 gives the comparison of the proposed architecture with the existing architectures in terms of Slice registers, Slice LUTs, Fully used LUT-FF pairs, Bonded IOBs. From the table, it is evident that the device utilization for the proposed design is lesser compared to the existing architectures.

Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice registers	1911	69120	2%
Number of Slice LUTs	3488	69120	5%
Number of fully used LUT-FF pairs	1574	3825	41%
Number of Bonded IOBs	423	640	66%
Number of Block RAM/FIFO	12	148	8%
Number of BUFG/BUFGCTRLs	1	32	3%

Table.1 Device Utilization Summary

Architecture	**Slice registers**	**Slice LUTs**	**Bonded IOBs**	**LUT-FF pairs**
Leelavathi.G et al., [14]	2635	4229	17	2252

Mahesh Walunjkar et al., [15]	2439	4229	17	2252
Yulin Zhang et al., [8]	2389	4401	388	2829
Ashwini R. Tonde et al., [16]	3699	6436	515	2338
Vedkiran Saini et al., [17]	10769	10558	389	NA
Proposed	1911	3488	423	1574

Table.2 Comparison of various architectures with proposed hybrid cryptosystem.

Also Fig. 9 represents the top module of the proposed design. It provides information regarding the inputs that are given to the module and the outputs that are obtained from the module. Xilinx ISE design suite is used for Verilog coding and ModelSim SE is used for simulation. The design is implemented on virtex-5 FPGA kit. Fig. 10 depicts the simulation results of the hybrid cryptosystem.

In Fig.10, the inputs to SHA-2 and encryption module and the corresponding outputs from the same modules are depicted. Also the key which is the output of SHA-2 module is obtained and given to encryption and decryption modules. Finally output from the decryption module is obtained which returns back the original plaintext or the data.

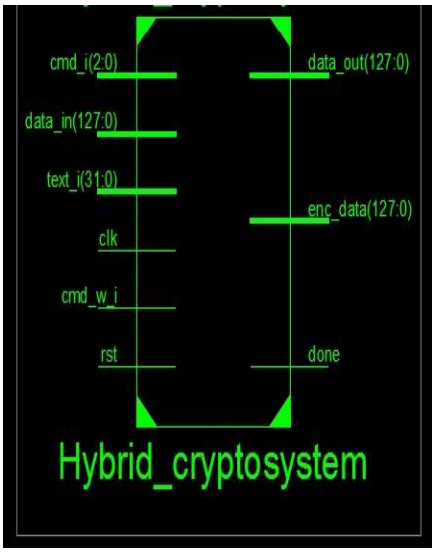

Fig. 9 Top module of hybrid cryptosystem

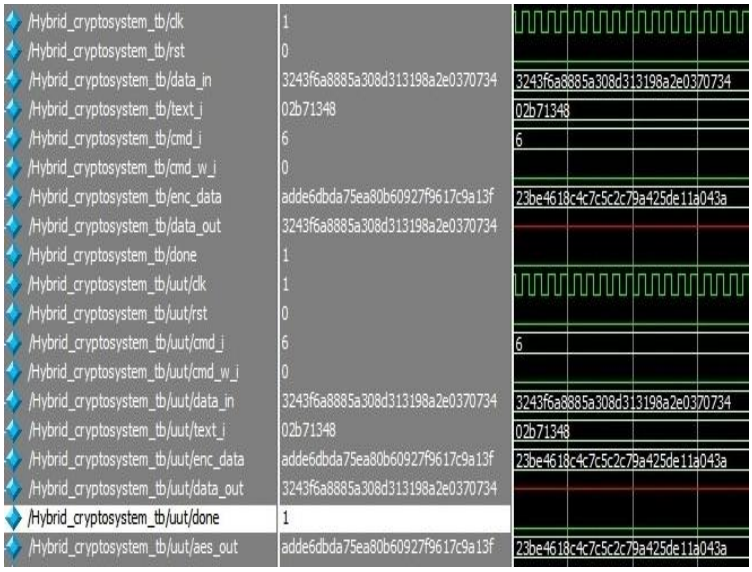

Fig. 10 Simulation result of Hybrid cryptosystem

VIII. Conclusion

In this paper, a hybrid cryptosystem is developed that integrates AES and SHA-2 algorithms. Since the complexity of the architecture is very high, higher data security is achieved. The design is synthesized using Xilinx ISE and implemented on Virtex-5 FPGA that consists of 110 million gates. The results obtained shows that the proposed design operates at a maximum frequency of 139.252 MHz with a delay of 7.181ns.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my guide Mrs. K N Pushpalatha, Associate Professor, Dept of ECE, Dayananda Sagar Institutions, for her valuable guidance throughout this work. Without her guidance and persistent help this work would have not been possible. I am very thankful to my parents and my husband (Sunny) for their immense love and trust on me throughout the journey of my life. I also thank all my M.Tech classmates & beloved friends for their perpetual support through all walks of my life.

REFERENCES

[1]. Jing Wang, Xiaoyang Zeng, Jun Chen, "A VLSI implementation of ECC combined with AES" 1-4244-0161 5/06/$20.00 ©2006 IEEE.

[2]. Behrouz,A.Forouzan, Debdeep Mukhopadhyay, "Cryptography and Network Security" 2nd Edition Tata McGraw Hill pvt ltd, New Delhi.

[3]. FIPS-197, NIST - National Institute of Standards and Technology, "Announcing the ADVANCED ENCRYPTION STANDARD (AES), "http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

[4]. N. SKLAVOS, O. KOUFOPAVLOU, "Implementation of the SHA-2 Hash Family Standard Using FPGAs" The Journal of Supercomputing, 31, 227–248, 2005 © 2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands.

[5]. Abhijith.P.S, Mallika Srivastava, Aparna Mishra, Manish Goswami, B.R.Singh, "High Performance Hardware Implementation of AES Using Minimal Resources" 2013 International Conference on Intelligent Systems and Signal Processing (ISSP).

[6]. Yasmin Alkady, Mohmed I. Habib, Rawya Y. Rizk,"A New Security Protocol Using Hybrid Cryptography Algorithms" 978-1-4799-3370-9/13/$31.00 ©2013 IEEE.

[7]. M.Meenakumari, G.Athisha, "Improving Message Authentication by Integrating Encryption with Hash function and its VLSI Implementation" International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering Vol. 2, Issue 1, January 2014.
[8]. Yulin Zhang, Xinggang Wang, "Pipelined Implementation of AES Encryption Based on FPGA" 978-1-4244-6943-7/10/$26.00 ©2010 IEEE.
[9].V. Sumathy & C. Navaneethan, "ENHANCED AES ALGORITHM FOR STRONG ENCRYPTION" International Journal of Advances in Engineering & Technology, Sept 2012. ©IJAET ISSN: 2231-1963.
[10]. Amandeep Kaur, Puneet Bhardwaj, Naveen Kumar, "FPGA Implementation of Efficient Hardware for the Advanced Encryption Standard" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-3, February 2013.
[11]. Adnan Abdul-Aziz Gutub, Farhan Abdul-Aziz Khan, "Hybrid Crypto Hardware Utilizing Symmetric-Key & Public-Key Cryptosystems" 2012 International Conference on Advanced Computer Science Applications and Technologies. 978-0-7695-4959-0/13 $25.00 © 2013 IEEE DOI 10.1109/ACSAT.2012.44.
[12]. Neeta Wadhwa, Syed Zeeshan Hussain and S.A.M Rizvi, "A Combined Method for Confidentiality, Integrity, Availability and Authentication (CMCIAA)" Proceedings of the World Congress on Engineering 2013 Vol II, WCE 2013, July 3 - 5, 2013, London, U.K. ISBN: 978-988-19252-8-2.
[13]. P.V.S. Shastry, A. Agnihotri, D. Kachhwaha, J. Singh and M.S. Sutaone, "A Combinational Logic Implementation of S-Box of AES," IEEE 54th Int. Midwest Symp on Circuits and Systems (MWSCAS), Aug. 2011, pp. 1-4.
[14]. Leelavathi.G, Prakasha S, Shaila K, Venugopal K R, L M Patnaik, "Design and Implementation of Advanced Encryption Algorithm with FPGA and ASIC" IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 3, June-July, 2013.
[15]. Mahesh Walunjkar, Md. Manan Mujahid, Syed Anwar Ahmed, Ashish Jadhav, "An AES-Core Development by Using Verilog" International Journal of Innovative Research in Computerand Communication Engineering, Vol. 1, Issue 8, October 2013.
[16]. Ashwini R. Tonde and Akshay P. Dhande, "Implementation of Advanced Encryption Standard (AES) Algorithm Based on FPGA" International Journal of Current Engineering and Technology E-ISSN 2277 – 4106, P-ISSN 2347 – 5161 ©2014 INPRESSCO®.
[17]. Vedkiran Saini, Parvinder Bangar, "Design and Implementation of Advanced Encryption Standard Algorithm-128 using Verilog" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-3, Issue-5, June 2014.

Author Biographies

K.N.Pushpalatha is working as an Associate Professor, Department of E&C, Dayananda Sagar College of Engineering, Bangalore. She received her B.E degree in Electronics& communication from Bangalore University and M.S. Degree in Electronics and Control from BITS Pilani. She is pursuing her Ph.d in Electronics at Mewar University, Rajasthan under the guidance of Dr. Arvind Kumar Gautham, Principal, S D College of Engineering, Muzzafarnagar, Uttara Pradesh. Her research interests include Image Processing, Biometrics, Information Theory and Coding and Signals and Systems. Contact: knpdrs@gmail.com

Vanishreeprasad S received her Bachelors of Engineering in the field of Medical Electronics in the year 2012 from Visvesvaraya Technological University. She started pursuing her Masters of Technology in the field of VLSI and Embedded Systems from the year 2013-2015. She has presented a paper in a National conference and she has been qualified in GATE 2012 with a GATE score of 362. Currently she is working towards her master's degree in VLSI and Embedded systems from Dayananda Sagar College of Engineering, Bangalore, India. Contact: vani2704@gmail.com.