TECH-CHRONICLE
AN INTERNATIONAL E-JOURNAL ON
EMERGING TRENDS IN SCIENCE, TECHNOLOGY AND MANAGEMENT

REVIEW ON FPGA IMPLEMENTATION OF IMAGE ENCRYPTION AND DECRYPTION USING AES ALGORITHM ALONG WITH KEY ENCRYPTION

Parul Rajoriya[1]
Department of E&TC
J.D. College of Engineering and Management

Nilesh Mohota[2]
Department of E&TC
J.D. College of Engineering and Management

Abstract— The importance of cryptography applied to security in electronic data transactions has acquired an essential relevance during the last few years. A proposed FPGA-based implementation of the Advanced Encryption Standard(AES) algorithm along with key encryption for images is presented in this paper. An efficient image encryption scheme based on FPGA using AES algorithm integrated with RC4 encryption standard is proposed. The use of RC4 algorithm imparts additional level of security to the encryption. The design converts the original image into its hex values using Matlab and then give it as input to the proposed AES. The key input given to AES is further encrypted using RC4 encryption algorithm. The encrypted image is decrypted using AES decryption algorithm. The encrypted key is decrypted using RC4 decryption algorithm so as to give it as input to the AES decryption algorithm. The encrypted and decrypted image can be analysed in Matlab. The design uses an iterative looping approach with block and key size of 128 bits, lookup table implementation of S-box. This gives low complexity architecture and easily achieves low latency as well as high throughput.

Keywords— AES algorithm, RC4 algorithm, image encryption, key encryption, image decryption, key decryption

I. INTRODUCTION

Each day millions of users generate and interchange large volumes of information in various fields, such as financial and legal files, medical reports, and bank services via Internet. These and other examples of applications deserve a special treatment from the security point of view, not only in the transport of such information but also in its storage. In this sense cryptography techniques are especially applicable.

For a long time, the Data Encryption Standard (DES) was considered as a standard for the symmetric key encryption. DES has a key length of 56 bits. However, this key length is currently considered small and can easily be broken. For this reason, the National Institute of Standards and Technology (NIST) opened a formal call for algorithms in September 1997. A group of fifteen AES candidate algorithms were announced in August 1998. Next, algorithms were subject to assessment process performed by various groups of cryptographic researchers all over the world. In August 2000, NIST selected five algorithms: Mars, RC6, Rijndael, Serpent and Twofish as the final competitors. These algorithms were subject to further analysis prior to the selection of the best algorithm for the AES. Finally, on October 2, 2000, NIST announced that the Rijndael algorithm was the winner.

Field Programmable Gate Arrays (FPGAs) are hardware devices whose function is not fixed which can be programmed in system. The potential advantage of encryption algorithm implemented in FPGAs includes:

Algorithm agility- This term refers to the switching of cryptographic algorithm during operation. Algorithm upload- It is perceivable that fielded devices upgraded with new encryption algorithm which did not exist at design time.

Algorithm modification- There are applications which require modification of a standardized algorithm. Architecture efficiency- With FPGAs it is possible to design and optimize architecture for specific parameter set.

Throughput- Although typically slower than ASIC implementation, FPGA have potential of running substantially faster than software implementations.

Cost efficiency- Time and cost for developing an FPGA implementation of a given algorithm are much lower than for an ASIC implementation.

As shown in Fig. 1 the cryptographic primitives are classified into three main categories; not using key, symmetric key and asymmetric key [1]. Symmetric key ciphers are also known as secret key or single key ciphers. Secret key ciphers are further classified as block ciphers and stream ciphers. In block ciphers, a block of bits/bytes is processed at a time. DES, IDEA, RC5, AES, BLOWFISH, TWOFISH are the different available block ciphers. Whereas in stream ciphers one bit or a byte of data is processed at a time. Stream ciphers are further classified as synchronous and self-synchronous stream ciphers.Different synchronous stream ciphers available in the literature are RC4, E0 (a stream cipher used in Bluetooth), A5/1 and A5/2 (stream ciphers used in GSM), SNOW 3G, ZUC (4G stream ciphers), Rabbit, FISH, and HC-256 etc. [2-7].

A keystream is produced in stream ciphers which is a pseudorandom sequence of bits. A plaintext (a sequence of bits/bytes) is converted into ciphertext (again a sequence of bits/bytes of same length as that of plaintext) by hiding the plaintext with a generated keystream, using a simple XOR operation. The strength of stream ciphers is a random keystream which ensures the computational security of the cipher.

In cryptography, the AES is also known as Rijndael. AES has a fixed block size of 128 bits and a key size of 128, 192 or 256 bits. This paper deals with an FPGA implementation of an AES encryptor/decryptor using an iterative looping approach with block and key size of 128 bits along with key encryption and decryption using RC4 algorithm . This method imparts additional level of security.

II. PREVIOUS WORK

In paper [8] Mazen El Maraghy et. Used AES-128 bit algorithm for optimization of area and speed. They have used 128 data bits as well as 128 bit cipher key. The implemented hardware design is evaluated in real time. M.Sambasiva Reddy et. [9] used the same AES-128 bit algorithm for speed, power consumption and area. They have implemented the AES algorithm using EDK. Hoang Trang et.[10]

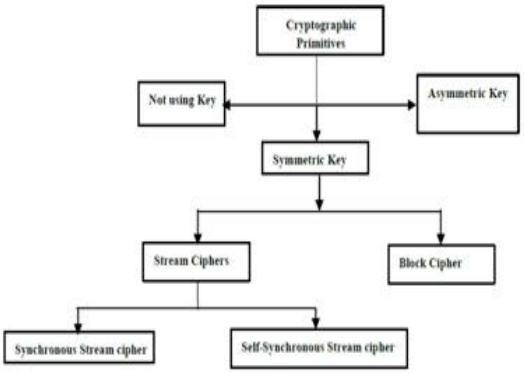

Fig. 1. Cryptographic primitives

This gives low complexity architecture and easily achieves low latency as well as high throughput. The design used an iterative looping approach with block and key size of 128 bit, lookup table implementation of S-box.Kamali S.H et. [11] used the modified advanced encryption algorithm to reflect a high level security and better image encryption.The modification is done by adjusting the ShiftRow Transformation.The author have compared the results of the previous AES algorithm and modified AES algorithm.

III. PROPOSED WORK

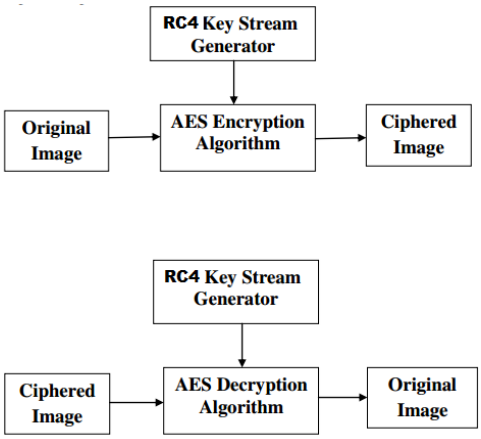

Fig 2. Proposed architecture

The figure 2 gives the proposed architecture in which image encryption and decryption is carried out using blend of AES algorithm and RC4 algorithm to provide additional level of security. The image encryption and decryption is carried out using AES algorithm and key encryption and decryption is carried out using RC4 algorithm.

A. AES ENCRYPTION AND DECRYPTION

The AES algorithm is a symmetric block cipher that can encrypt and decrypt information. Encryption converts data to an unintelligible form called cipher-text. Ecryption of the cipher-text converts the data back into its original form, which is called plain-text.

a. AES Encryption

The AES algorithm operates on a 128-bit block of data and executed Nr - 1 loop times. The key length is 128, 192 or 256 bits in length respectively. The first and last rounds differ from other rounds in that there is an additional AddRoundKey transformation at the beginning of the first round and no MixCoulmns transformation is performed in the last round. In this paper, we use the key length of 128 bits (AES-128) as a model for general explanation. An outline of AES encryption is given in Fig. 3a)

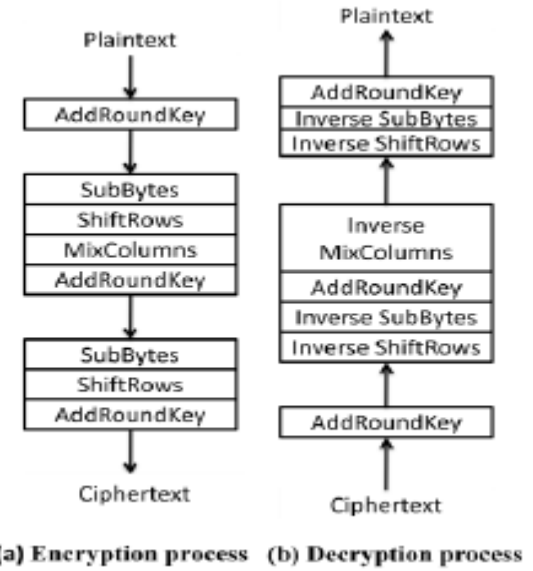

Fig 3. AES Encryption and Decryption process

SubBytes Transformation:

The SubBytes transformation is a non-linear byte substitution, operating on each of the state bytes independently. The SubBytes transformation is done using a once-precalculated substitution table called S-box. That S-box table contains 256 numbers (from 0 to 255) and their corresponding resulting values. This approach has the significant advantage of performing the S-box computation in a single clock cycle, thus reducing the latency and avoids complexity of hardware implementation.

ShiftRows Transformation:

In ShiftRows transformation, the rows of the state are cyclically left shifted over different offsets. Row 0 is not shifted; row 1 is shifted one byte to the left; row 2 is shifted two bytes to the left and row 3 is shifted three bytes to the left.

MixColumns Transformation:

In MixColumns transformation, the columns of the state are considered as polynomials over GF (28) and multiplied by modulo x4 + 1 with a fixed polynomial c(x), given by: c(x)={03}x3 + {01}x2 + {01}x + {02}.

AddRoundKey Transformation:

In the AddRoundKey transformation, a Round Key is added to the State - resulted from the operation of the MixColumns transformation

- by a simple bitwise XOR operation. The RoundKey of each round is derived from the main key using the KeyExpansion algorithm. The encryption/ decryption algorithm needs eleven 128-bit RoundKey, which are denoted RoundKey(0) RoundKey(10).

b. AES Decryption

Decryption is a reverse of encryption which inverse round transformations to computes out the original plaintext of an encrypted cipher-text in reverse order shown in fig.3.b). The round transformation of decryption uses the functions AddRoundKey, InvMixColumns, InvShiftRows, and InvSubBytes successively.

AddRoundKey:

AddRoundKey is its own inverse function because the XOR function is its own inverse. The round keys have to be selected in reverse order.

InvShiftRows Transformation:

InvShiftRows exactly functions the same as ShiftRows, only in the opposite direction. The first row is not shifted, while the second, third and fourth rows are shifted right by one, two and three bytes respectively.

InvSubBytes transformation:

The InvSubBytes transformation is done using a onceprecalculated substitution table called InvS-box. That InvS-box table contains 256 numbers (from 0 to 255) and their corresponding values.

InvMixColumns Transformation:

The InvMixColumns transformation is done using polynomials of degree less than 4 over GF(28), which coefficients are the elements in the columns of the state, are multiplied modulo ($x4 + 1$) by a fixed polynomial $d(x) = \{0B\}x3 + \{0D\}x2 + \{09\}x + \{0E\}$, where $\{0B\}$, $\{0D\}$; $\{09\}$, $\{0E\}$ denote hexadecimal values.

B. RC4 ENCRYPTION AND DECRYPTION

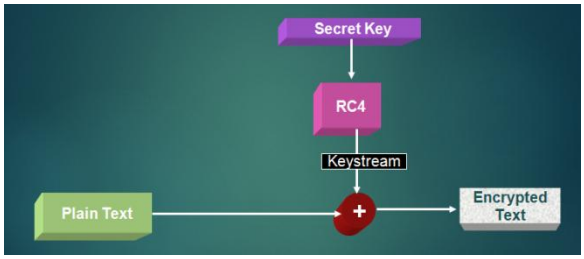

Fig 4. Block diagram of RC4 algorithm

RC4 follows the design strategy used in stream ciphers. To extract the pseudorandom data bytes from a pseudorandom permutation is the basic design principle of RC4 stream cipher. RC4 has two working modules: first there is a KSA with key K as input (with typical size of 40-256 bits), and second is PRGA which generates a pseudo-random output sequence. The pseudo code for RC4. Fig 4 presents the complete working of RC4 encryption algorithm. KSA generates the 256 byte initial state vector S, by scrambling input state vector with a random key *K*. The S contains a permutation of 8 bit words i.e. 256 bytes. The secret key *k* is generally of length between 8 to 2048 bits and the expanded key K (K of length N=256 bytes) is produced by performing simple repetitions. The expanded key is generated in the manner such that if secret key *k* is of length *l* bytes, the expanded key will be K[i] = *k* [*i* mod *l*] for $0 \le i \le N-1$. Further *S* pairs are swapped and an initial state SN-1 is achieved at the end which is the input to the second module PRGA. It generates the keystream of words and is further XORed with the plaintext to produce a ciphertext. Fig 5 depicts both the modules.

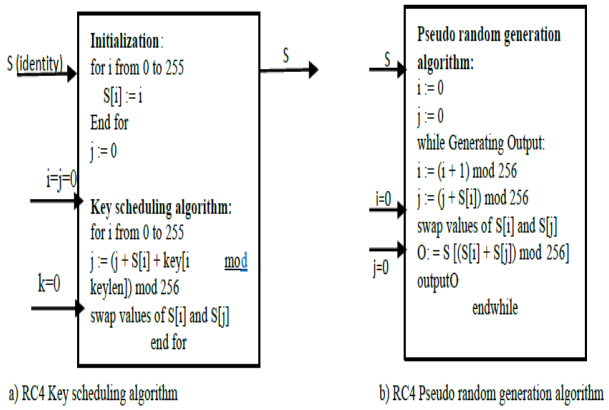

Fig 5.RC4 Stream Cipher

C. DESIGN STEPS

The data block is processed as follows:

i. The AES encryption routine begins by copying the 16-byte input array into a 4×4 byte matrix.
ii. Input Image block is XOR ed with the first 128-bits of the cipher key which is the output of the RC4 encryption algorithm.
iii. Then the resulting matrix is serially passed through 10 rounds.
iv. The result of the last round is encrypted image.

REFERENCES

[1]Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Hand-book of Applied Cryptography. CRC Press, August 2011 edition, 1996. Fifth Printing.

[2]Alex Biryukov, Adi Shamir, and David Wagner. Real time cryptanalysis of A5/1 on a PC. In Bruce Schneier, editor, *FSE*, volume 1978 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2000.

[3] Bluetooth T M. Bluetooth specification, v4.0, June 2010. E0 encryption algorithm described in volume 2, pages 1072–1081. Available online at http://www.bluetooth.org.

[4] Marc Briceno, Ian Goldberg, and David Wagner. A pedagogical implementation of the GSM A5/1 and A5/2 "voice privacy" encryption algorithms. Available online at http://www.scard.org/gsm/a51.html, 1998.

[5] 3rd Generation Partnership Project. Specification of the 3GPP confidentiality and integrity algorithms UEA2 & UIA2. ETSI/SAGE Specification Document 2: SNOW 3G Specification, v1.1, September 6, 2006.

[6] ECRYPT Stream Cipher Project eSTREAM. The current eSTREAM portfolio. Available online at http://www.ecrypt.eu.org/stream/ index.html.

[7] ECRYPT Stream Cipher Project eSTREAM. Software performance results from the eSTREAM project. Available online at http://www. ecrypt.eu.org/stream/perf/#results.

[8] El. Maraghy M, Hesham S and Abd El. Ghany M.A, "Real-time Efficient FPGA implementation of AES algorithm", IEEE International SOC Conference(SOCC), page 203-208, Sept 2013.

[9] M.Sambasiva Reddy and Mr.Y.Amar Babu, "Evaluation Of Microblaze and Implementation Of AES Algorithm using Spartan-3E", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 7, page 3341-3347, July 2013.

[10] Hoang Trang and Nguyen Van Loi, "An Efficient FPGA Implementation of The Advanced Encryption Standard algorithm", IEEE International Conference on Computing and Communication Technology, page 1-4, Ho Chi Minh city, 2012.
[11]Kamali S.H, Shakerian R, Hedayati M and Rahmani M, "A new modied version of Advanced Encryption Standard based algorithm for image encryption", (ICEIE) International Conference On Electronics and Information Engineering, volume 1, page 1250-1255, Aug 2010.