ISSN 2319 – 2518 www.ijeetc.com
Vol. 2, No. 4, October 2013
© 2013 IJEETC. All Rights Reserved

Research Paper

DESIGN AND IMPLEMENTATION OF A PRIVATE AND PUBLIC KEY CRYPTO PROCESSOR

Swati G Mavinkattimath[1]* and N S Sirdeshpande[2]

*Corresponding Author: Swati G Mavinkattimath, ✉ swatisgm@gmail.com

This paper presents the design and implementation of a crypto processor, a special-purpose microprocessor optimized for the execution of cryptography algorithms. This crypto processor can be used for various security applications such as storage devices, embedded systems, network routers, security gateways using IPSec and SSL protocol, etc. The crypto processor consists of coprocessor blocks dedicated to the AES, triple-DES private key crypto algorithms and RSA public key crypto algorithm. The dedicated coprocessor block permits fast execution of encryption, decryption, and key scheduling operations. The crypto processor has been designed and implemented using an MATLAB.

Keywords: Encryption/Decryption, Data encryption standard, Advanced encryption standard, RSA algorithm, MATLAB, Public key, Private key

INTRODUCTION

Nowadays, nearly all companies, government agencies and home users depend on computer systems and communication systems such as the Internet and Intranet. The advent of computers and networks has completely changed the way in which we live and work. The expansion of the worldwide communication network such as the Internet and the increased dependency on digitized information in our society makes information more vulnerable to abuse. If there are security problems in these information systems, users will fear that their sensitive information may be monitored and business secrets stolen. For these reasons, it is important to make information systems secure by protecting data and resources from malicious acts crypto (cryptography) algorithms are the core of such security systems.

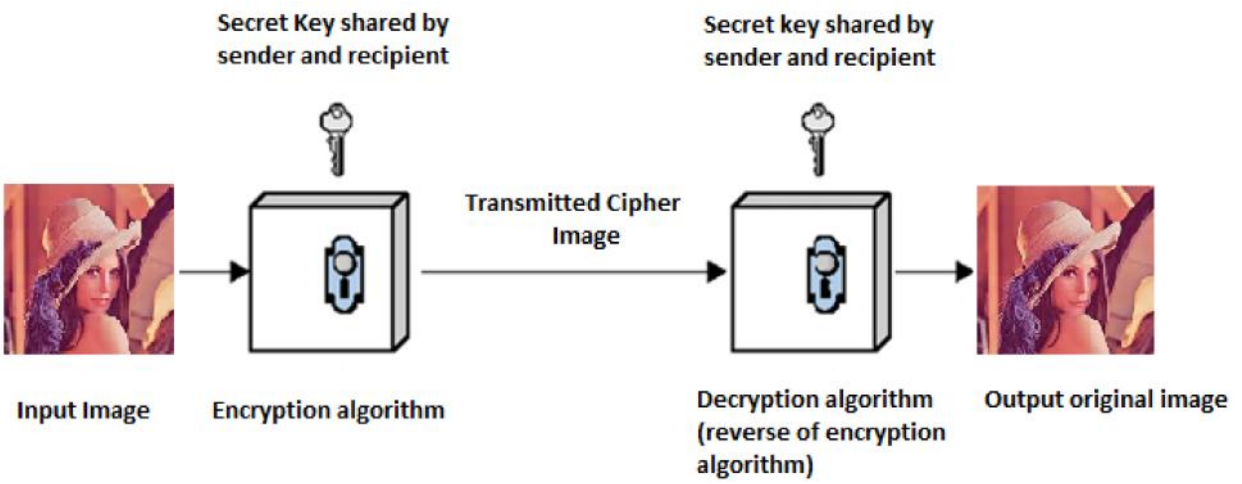
Cryptography (Encryption/ Decryption)

Cryptography is usually referred to as the study of secret, while now days it is most attached to the definition of encryption. Encryption is the process of converting plain text to a cryptic text

[1] Department of Electronics and Communication, KLE Dr. M S Sheshgiri College of Engineering and Technology, Belgaum.

[2] Department of Electronics and Communication, K.L.S's Gogte Institute of Technology, Belgaum.

Figure 1: Encryption / Decryption


Secret Key shared by sender and recipient
Secret key shared by sender and recipient
Transmitted Cipher Image
Input Image
Encryption algorithm
Decryption algorithm (reverse of encryption algorithm)
Output original image

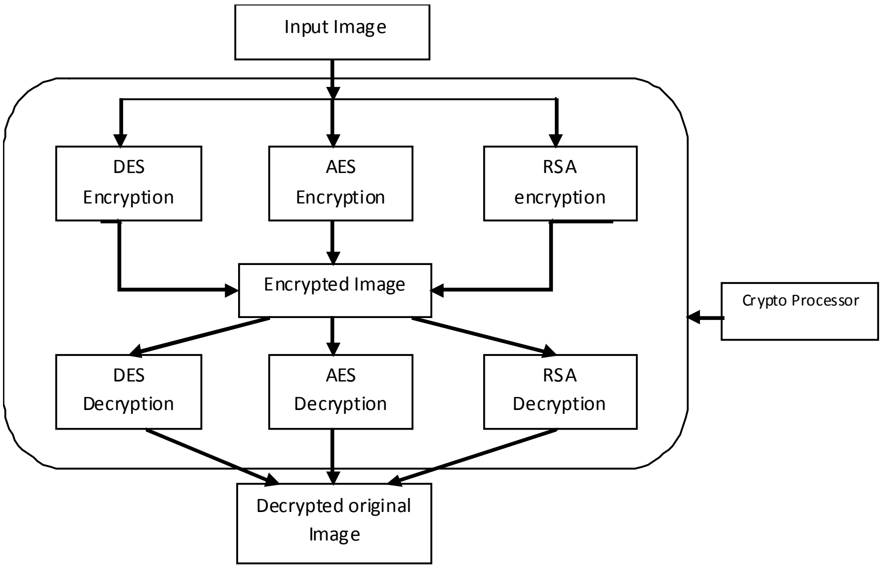
to secure it against data thieves. This process has another part where cryptic text needs to be decrypted on the other end to be understood in Figure 1.

METHODOLOGY

The Crypto-Processor Architecture

- The Crypto-Processor Architecture (HoWon and Sunggu, 2004) is as shown in Figure 2.
- The crypto processor consists of three dedicated blocks for private and public key cryptography.
- Data encryption Standard and Advanced encryption Standard crypto blocks are used for Private Key encryption and RSA crypto block is used for Public Key cryptography.
- This crypto-processor can be interfaced with the 32-bit RISC type crypto controller

Figure 2: Block Diagram of Crypto Processor



that controls the dedicated crypto block and can perform the interface operations with external devices such as memory and an I/O bus interface controller.

- The dedicated crypto block results in fast execution of the encryption, decryption and key scheduling operations for the AES and triple-DES algorithms and enables fast scalar multiplication and exponentiation operations for the crypto algorithms.

Crypto Block for the Private Key Crypto Algorithm

Data Encryption Standard Crypto Block

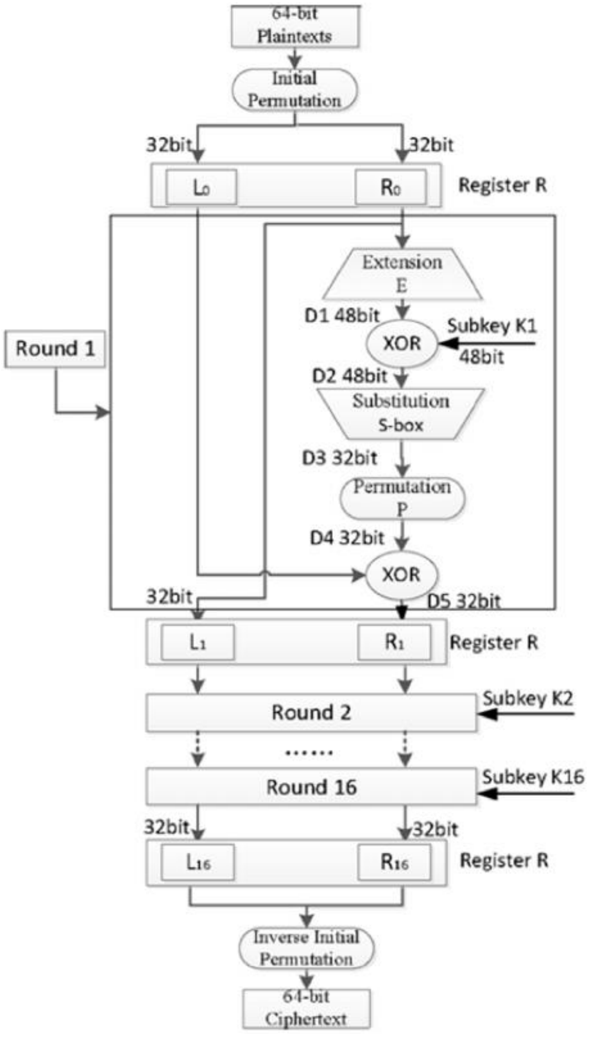
Without doubt the first and the most significant modern symmetric encryption algorithm is the Data Encryption Standard (DES).The DES was published by the United States National Bureau of Standards in January 1977 as an algorithm to be used for unclassified data (information not concerned with national security). The Data Encryption Standard (DES) is a block cipher operating on 64-bit data blocks. The encryption transformation depends on a 56-bit secret key and consists of sixteen Feistel iterations surrounded by two permutation layers.

- An initial bit permutation *IP* at the input
- An inverse IP^{-1} at the output.

The structure of the cipher is depicted in Figure 3. The decryption process is the same as the encryption, except for the order of the round keys used in the Feistel iterations (Chong, 2012).

The 16-round Feistel network, which constitutes the cryptographic core of DES, splits the 64-bit data blocks into two 32-bit words, LBlock and RBlock (denoted by L_0 and R_0). In each iteration (or round), the second word R_i is fed to a function f and the result is added to the first word L_i. Then both words are swapped and the algorithm proceeds to the next iteration. The function *f* of DES algorithm is key dependent and consists of 4 stages.

Figure 3: Data Encryption Standard Sequence



DES (Data Encryption Standard) is a block cipher which uses a 64-bit key and operates on 64-bit blocks of data. DES has a 56-bit key, because every 8th bit of the 64-bit key is used for parity checking. In the DES algorithm, there are 16 rounds of identical operations such as non-linear substitutions and permutations. In each round, 48-bit sub-keys are generated, and substitutions using S-box, bitwise shift, and XOR operations are performed.

DES Decryption

The decryption process with DES is essentially the same as the encryption process and is as follows:

Use the input to the DES algorithm but use the keys Ki in reverse order. That is, use K16 on the first iteration, K15 on the second until K1 which is used on the 16th and last iteration.

Advanced Encryption Standard Crypto Block: In September of 1997 the National Institute of Standards and Technology (NIST) issued a request for possible candidates for a new Advanced Encryption Standard (AES) to replace the DES. Then in October of 2000, NIST announced the cipher Rijndael, which is developed by Joan Daemen and Vincent Rijmen, as an AES algorithm. Rijndael is a block cipher using 128, 192, 256-bit input/output and keys. The sizes of data blocks and keys can be chosen independently (Mehran and Arash, 2012). With regard to using a key length other than 128 bits, the main thing that changes in AES is how you generate the key schedule from the key.

Encryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Except for the last round in each case, all other rounds are identical. Each round of processing includes one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key.

The order in which these four steps are executed is different for encryption and decryption. To appreciate the processing steps used in a single round, it is best to think of a 128-bit block as consisting of a 4 × 4 matrix of bytes, arranged as follows:

$$\begin{bmatrix} byte0 & byte4 & byte8 & byte12 \\ byte1 & byte5 & byte9 & byte13 \\ byte2 & byte6 & byte10 & byte14 \\ byte3 & byte7 & byte11 & byte15 \end{bmatrix}$$

The output state array produced by the last round is rearranged into a 128-bit output block. Unlike DES, the decryption algorithm differs substantially from the encryption algorithm. Although, overall, the same steps are used in encryption and decryption, the order in which the steps are carried out is different.

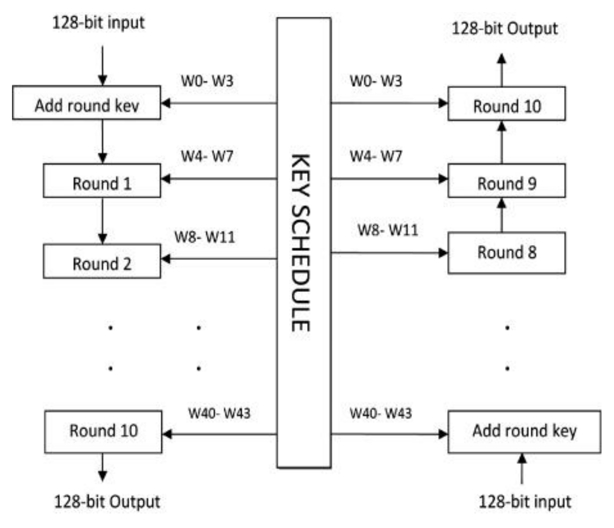
Each round of processing in AES involves byte-level substitutions followed by word-level permutations. Speaking generally, DES also involves substitutions and permutations, except that the permutations are based on the Feistel notion of dividing the input block into two halves, processing each half separately, and then swapping the two halves. The Advanced Encryption Standard is as shown in Figure 4.

Crypto Block for the Public Key Crypto Algorithm

RSA Crypto Block

The RSA is a public key cryptographic algorithm that is used to help ensure data communication security. It is simply based on two main cryptographic processes. First, using a public key it converts an input data called the plaintext into an unrecognizable encrypted output called cipher text (encryption process),

Figure 4: AES Encryption / Decryption



such that it is impossible to recover the original plaintext without the encryption password in a reasonable amount of time. Second, using a private key, the RSA then converts the unrecognizable data back to its original form (decryption process). Today it is used in web browsers, email programs, mobile phones, virtual private networks and secure shells. Until recently, the use of RSA was very much restricted by patent and export laws. However, the patent has now expired and US export laws have been relaxed.

The RSA Model Description

The RSA model is shown in Figure 5.

The step by step process of the RSA algorithm is as follows

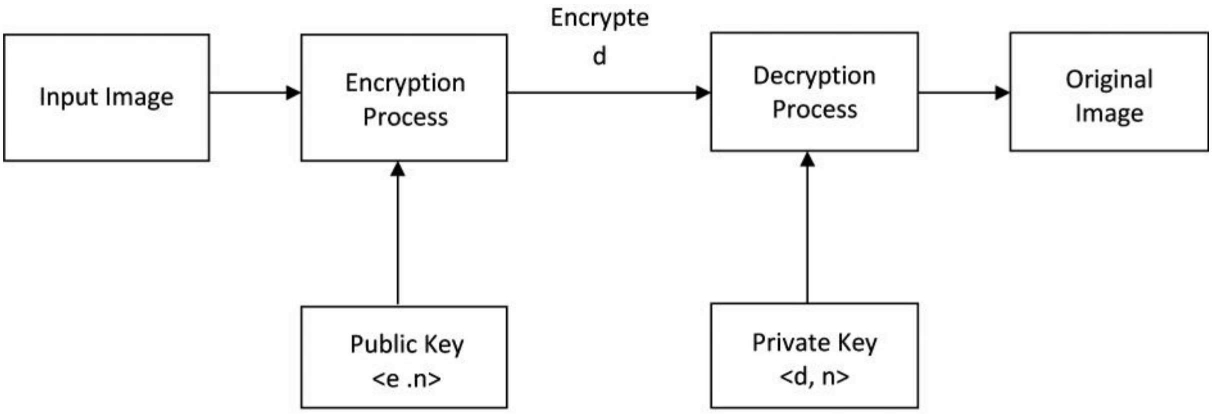
- Generate a public key (e) and a private key (d) by choosing two very large prime numbers p and q each around 256 bits (75 digits).
- Multiply p and q and let the result be n.

  $n = p * q$.

  Note that the factors p and q remain secret and n is public. Even if n is known it is not practically feasible to get back p and q since factorizing a very large number is computationally infeasible.
- Generate a public key by choosing a number e, which is relatively prime to the "totient" function

  ¨(n) = (p – 1) (q – 1)
- Generate a private key by choosing a number d, which is the multiplicative inverse of e mod ¨(n). Note that the public key is <e, n> which is known to everyone and the private key is <d, n> which is known only to

Figure 5: RSA Model



the person who has to decrypt or sign the message.

- Encrypt a message $m(<n)$, raise m to the power e under modulo n. The result is the cipher text (c).

  $c = m^e \mod n$

- Decrypt the cipher text, raise the cipher to the power d under modulo n.

  $m = c^d \mod n$

- Sign the message by encrypting it with the private key $<d, n>$ and decrypting it with the public key $<e, n>$.

RESULTS AND DISCUSSION

In this paper, the following factors are used such as the Key length value; Simulation speed, the key length management, the encryption ratio, power consumption, scalability, key used and the security of data against attacks are discussed in Table 1.

- Developed: It states about the timeline of algorithm.
- Key Length Value: It plays a vital role that shows how data is encrypted.
- Type of Algorithm: Two type of algorithm exist. Based on process and key it is segregated as symmetric and asymmetric.
- Encryption Ratio: Measures amount of data that is to be encrypted. It should be minimized to reduce complexity. In our analysis we stated three levels like low, medium, high.
- Security Issues: Encryption technique must satisfy cryptographic security like plaintext-cipher text attack.
- Simulation Speed: Encryption and Decryption algorithms are fast enough to meet real time requirements.
- Scalability: Key size and block size variation is referred as scalability.
- Key Used: To specify whether same key is used for encryption and decryption process or different key.

Table 1: Comparison of DES, AES and RSA Algorithm

S. No.	Factor: Analysed	DES	AES	RSA
1.	Developed	1977	2000	1978
2	Key length value	138,192, 256 bits	56 bits	>1024 bits
3.	Type of algorithm	Symmetric	Symmetric	Asymmetric
4.	Encryption Ratio	Low	High	High
5.	Security attacks	Inadequate	Highly secured	Timing secure
6.	Stimulation speed	Fast	Fast	Fast
7.	Scalability	Scalable algorithms	No scalability occurs	No scalability occurs
8.	Key used	Same key used for encrypt and decrypt process	Different key used for encrypt and decrypt process	Different key used for encrypt and decrypt process
9.	Power consumption	Low	Low	High
10.	Hardware and Software implementation	Better in hardware than in software	Faster and efficient	Not very efficient

- Power Consumption: Measure the power in units when the process takes place. It stated in two levels such as high and low.
- Implementation: Hardware and Software are effective in AES compared to DES and RSA.

The Results of simulation in MATLAB of DES, AES and RSA are shown in Figures 6, 7 and 8. The encrypted image is the output of encryption algorithm which is scrambled and the output of decryption algorithm is same as original image.

Figure 6: Experiment Output of AES

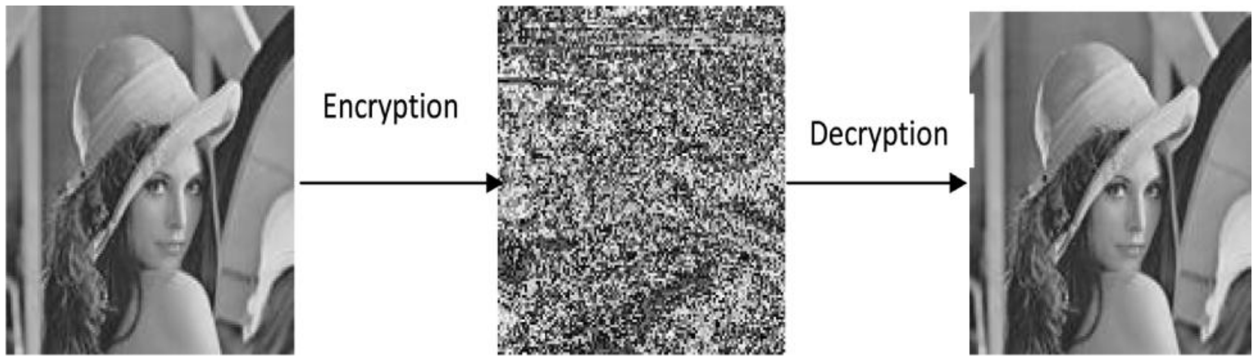


Figure 7: Experiment Output of DES

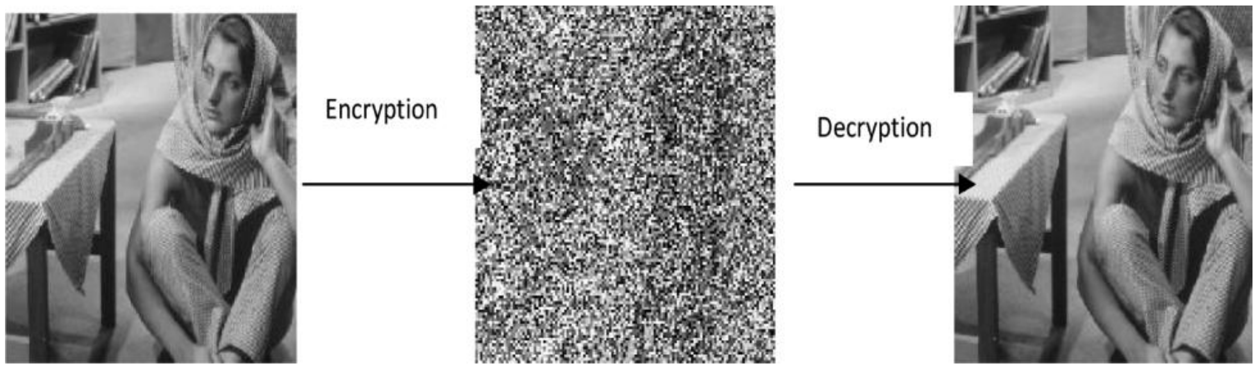
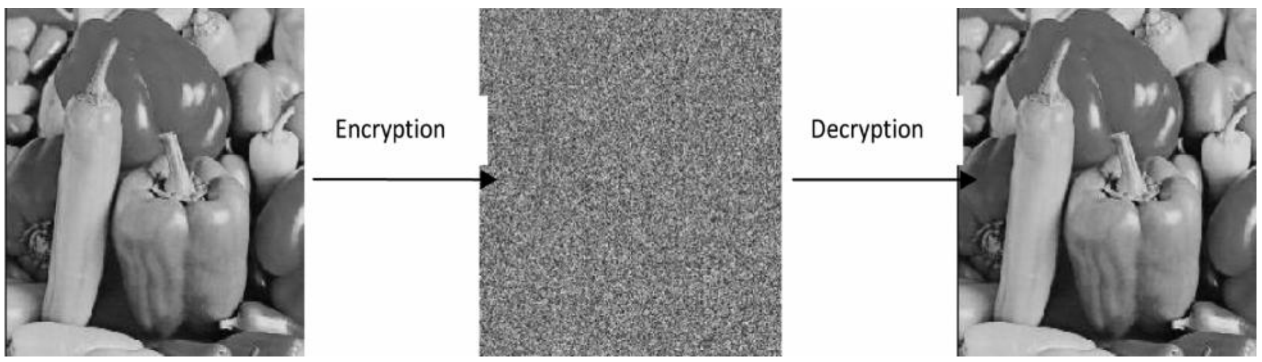


Figure 8: Experiment Output of RSA



CONCLUSION

In Data communication, encryption algorithm plays an important role. In this work different public and private key crypto algorithms have been studied and implemented in MATLAB with Lena image. Those encryption techniques are studied and analyzed well to promote the performance of the encryption methods also to ensure the security. It is seen from the results that the decrypted image is same as input image in all algorithm techniques. However the decryption of AES algorithm is better than other algorithms. From the simulation result, we evaluated that AES algorithm is much better than DES and RSA algorithm. The dedicated block of the crypto processor accelerates private and public key crypto algorithms and the programmability of the crypto controller makes possible fast execution of various security applications.

SCOPE FOR FUTURE WORK

We have implemented crypto blocks in a crypto processor with three dedicated blocks of AES, DES and RSA. The further study may be carried out by:

- Implementing more private and public key Crypto blocks.
- Interfacing crypto processor with host processor along with controller to control the operations of crypto processor without interrupting host processor.

REFERENCES

1. Bruce Schneier (1996), *Applied Cryptography*, 2nd Edition, John Wiley & Sons, Inc., New York.
2. Chandra M Kota and Cherif Aissi (2002), "Implementation of the RSA Algorithm and its Cryptanalysis", University of Louisiana at Lafayette, College of Engineering Lafayette, LA 70504, USA.
3. Chong Hee Kim (2012), "Improved Differential Fault Analysis on AES Key Schedule", *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 1, February.
4. Guang Gong and Solomon W Golomb (1999), "Transform Domain Analysis of DES", *Fellow, IEEE*.
5. HoWon Kim and Sunggu Lee (2004), "Design and Implementation of a Private and Public Key Crypto Processor", Vol. 50, No. 1, February.
6. Kun Ma, Han Liang and Kaijie Wu (2012), "Homomorphic Property-Based Concurrent Error Detection of RSA: A Countermeasure to Fault Attack", *IEEE Transactions on Computers*, Vol. 61, No. 7, July.
7. Mao-Yin Wang, Chih-Pin Su, Chia-Lung Horng, Cheng-Wen Wu and Chih-Tsun Huang (2010), "Single-and Multi-Core Configurable AES Architectures for Flexible Security", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 18, No. 4, April.
8. Mehran Mozaffari-Kermani and Arash Reyhani-Masoleh (2012), "Efficient and High-Performance Parallel Hardware Architectures for the AES-GC", *IEEE Transactions on Computers*, Vol. 61, No. 8, August.
9. National Institute of Standards and Technology (December 1993), FIPS Publication 46-2: Data Encryption

Standard, National Institute for Standards and Technology, Gaithersburg, MD, USA.

10. Salah Zaher, Amr Badr and Ibrahim Farag (2012), "Performance Enhancement of RSA Cryptography Algorithm by Membrane Computing", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol. 2, No. 9, September.

11. Scheier B (1995), *Applied Cryptography: Protocols, Algorithms and Source Code in C*, 2nd Edition, John Wiley & Sons.

12. Shah Kruti R and Bhavika Gambhava (2012), "New Approach of Data Encryption Standard Algorithm", *International Journal of Soft Computing and Engineering (IJSCE)*, Vol. 2, No. 1, March, ISSN: 2231-2307.

13. Weiwei Shan, Xin Chen, Bo Li, Peng Cao, Jie Li, Gugang Gao and Longxing Shi (2013), "Evaluation of Correlation Power Analysis Resistance and Its Application on Asymmetric Mask Protected Data Encryption Standard Hardware".