Journal of Computer Science 10 (9): 1752-1757, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.1752.1757 Published Online 10 (9) 2014 (http://www.thescipub.com/jcs.toc)

A NOVEL SUBSTITUTION BOX DESIGN FOR HUMMING BIRD-2 AGAINST SIDE CHANNEL ATTACK

[1]Blesslin Sheeba, T. and [2]P. Rangarajan

[1]Department of ECE, Sathyabama University, Chennai-600087, India
[2]Department of EEE, RMD Engineering College, Chennai-600087, India

Received 2014-02-13; Revised 2014-04-17; Accepted 2014-04-23

ABSTRACT

The side-channel attacks are one of the effortless and commanding attacks against cryptographic implementation and their intention vary from protocols, modules, primitives and system. As a result of this attack, a serious threat to the security of cryptographic module was encountered. In effect, realization of the cryptographic algorithm has to take some countermeasure to resist against this type of attacks. This study presents an efficient S-Box design using Null Conventional Logic (NCL) for Humming Bird-2 which is an ultra lightweight cryptographic algorithm. The recommended architecture was developed using Verilog HDL and implemented on altera cyclone IV E. The results are promising in terms of Area and Degree of Confusion (DOC).

Keywords: Null Conventional Logic (NCL), Substitution Box (S-Box), Degree of Confusion (DOC), Cryptography, Humming Bird-2, Side Channel Attack (SCA)

1. INTRODUCTION

The increased enslavement on digitized information in our society and the expansion of worldwide communication network like internet makes information more vulnerable to intruder *Takemura*. If the information suffers from security problem the consumer will fear that their business secrets were stolen and sensitive information may be monitored. This is the reason were cryptographic algorithm evolved; hence the valuable information and data are protected from the malicious act. The target specific objective has been saved from the hackers by constructing a security mechanism, which includes cryptographic algorithm like symmetric cipher, public-key cipher and hash functions *Shivkumar etal*. But in practice this prevention is far from the complete security solutions, since it is nature that the attacker will not only depend on the computational complexity to break the cipher employed in security systems *Aris etal*.

In reality, the realization of cryptographic cipher is foot on physical devices (hardware or software) which intermingle with and influenced by their settings. The information is valuable for cryptanalysis which is abstracted from that physical medium interaction *Raphael etal*. This type of information is called as side-channel information and attack based on this data is called as side-channel attack. This attack rely on the way, the cryptographic algorithm is implemented, somewhat than the algorithm itself. In tradition, the cryptanalysis is purely based on mathematical object hence the attack rely on side-channel information is also called as Implementation attack *Rajakumar etal*. In 1965 the first and official information for the SCA attack have been revealed. The attack is based on power consumption and timing computation.

Figure 1 shows the conventional cryptographic model including side channel attacks. The SCA is based on two types namely software and hardware. In hardware, SCA is achieved by the physical parameters like Electromagnetic Radiation, heat, sound, visible light, power consumption, execution time, frequency, error messages and faulty outputs *Salem etal*. Among which electromagnetic radiation, execution time and power consumption are some famous attacks *Faudzi etal*.

Corresponding Author: Blesslin Sheeba, T., Department of ECE, Sathyabama University, Chennai-600087, India

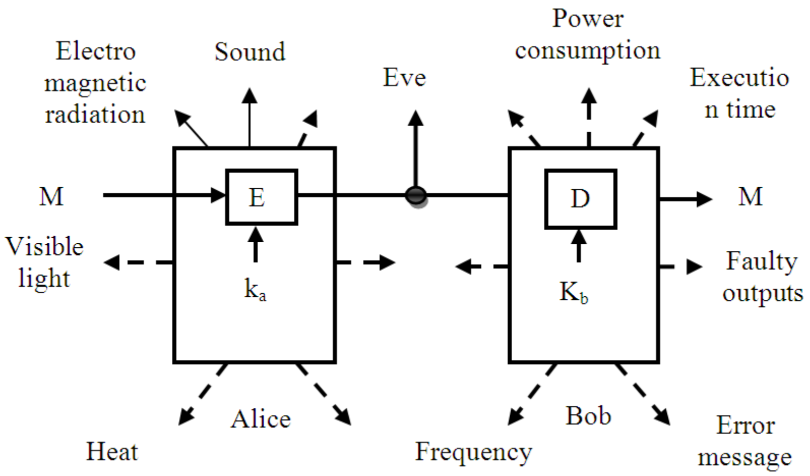

Fig. 1. Conventional cryptographic model with SCA

On the other hand, the software based SCA attack rely on Mathematical calculation, key breaking, S-Box intrusion and reverse-engineering; out of which S-Box implementation is research of interest now. From here onwards, the paper will discuss the SCA attack based on the S-Box intrusion, which is an attack based on software realization.

Design of S-Box falls under two circumstances which is Boolean Function Representation (BFR) and Hardware-Friendly *Rohini etal*. In BFR, the Boolean function involved in S-Box design should be optimized with high data security, so that the Timing attack of the data can be prevented. In Hardware-Friendly design, the S-Box architecture is optimized so that the area is consumed in the target hardware platform in which it is implemented *Abdelkrim etal.*

This study presents an S-Box design based on Null Conversion Logic (NCL) which shows a remarkable performance improvements in encrypted data based on Degree of Confusion. The rest of the paper proceed as follow, Humming Bird-2 (HB-2) algorithm, NCL based S-Box, Performance metrics and Conclusion.

2. CRYPTOGRAPHIC ALGORITHM

2.1. Humming Bird-2

The algorithm for Encryption and Decryption of Humming Bird-2 is shown below:

Algorithm: *Hummingbird Encryption*
Input: *A 16-bit Plain text PT_t and*
four rotors RSi_t (i =1, 2, 3, 4)
Output: *A 16-bit cipher text CT_t*

```
$V12_t = E_{k1} (PT_i \boxplus RS1_t)$ [Block encryption]
$V23_t = E_{k2} (V12_t \boxplus RS2_t)$
$V34_t = E_{k3} (V23_t \boxplus RS3_t)$
$CT_i = E_{k4} (V34_t \boxplus RS4_t)$
$LFSR_t +1 \leftarrow LFSR$ [State updating]
$RS_{1t+1} = RS_{1t} \boxplus V34_t$
$RS3_{t+1} = RS3_t \boxplus V23_t \boxplus LFSR_{t+1}$
$RS4_{t+1} = RS4_t \boxplus V12_t \boxplus RS1_{t+1}$
$RS2_{t+1} = RS2_t \boxplus V12_t \boxplus RS4_{t+1}$
return $CT_i$
```

Algorithm: *Hummingbird Decryption*
Input: *A 16-bit cipher text CT_i and*
four rotors RSi_t (i =1, 2, 3, 4)
Output: *A 16-bit Plain text PT_i*

```
$V34_t = D_{k4} (CT_i) \boxplus RS4_t$[Block decryption]
$V23_t = D_{k3} (V34_t) \boxplus RS3_t$
$V12_t = D_{k3} (V23_t) \boxplus RS2_t$
$PT_i = D_{k1} (V12_t) \boxplus RS1_t$
LFSRt+1 $\leftarrow$ LFSR[State updating]
$RS_{1t+1} = RS_{1t} \boxplus V34_t$
$RS3_{t+1} = RS3_t \boxplus V23_t \boxplus LFSR_{t+1}$
$RS4_{t+1} = RS4_t \boxplus V12_t \boxplus RS1_{t+1}$
$RS2_{t+1} = RS2_t \boxplus V12_t \boxplus RS4_{t+1}$
return $PT_i$
```

It has 128-bit key and 128-bit internal state register R which is initialized using vector IV (64-bit). The operation involved in HB-2 were exclusive-OR, 65536 addition modulo, non linear mixing function f(x) and subtraction modulo which are performed on 16-bit as shown in **Fig. 2.**

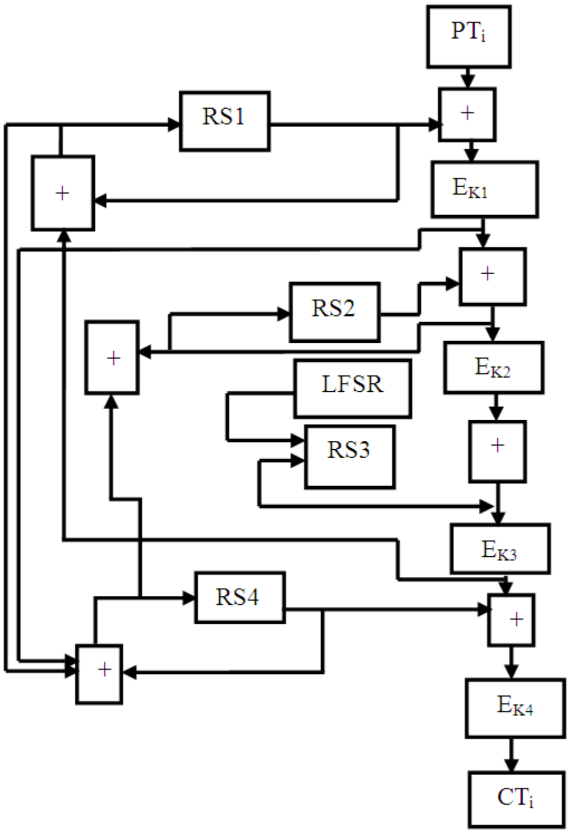

Fig. 2. Encryption for HB-2

The below calculation will compute the non linear mixing function f(x):

$$S(x) = S_1(x_0)|S_2(x_1)|S_3(x_2)|S_4(x_3)$$

$$L(x) = x \oplus (x <<< 6) \oplus (x <<< 10)$$

$$f(x) = L(S(x))$$

where, linear transformation is denoted by L(x) and computation of four S-Boxes is denoted by S(x).

The 16-bit keyed permutation is computed using the following expression:

$$\text{WD16}\ (x, a, b, c, d) = f(f(f(f(x \oplus a) \oplus b) \oplus c) \oplus d)$$

The internal state of Humming Bird-2 is initialized using a four round computation; the decryption algorithm is vice versa of encryption process which is shown in **Fig. 3:**

$$R^{(0)} = (IV_1, IV_2, IV_3, IV_4,\ IV_1, IV_2, IV_3, IV_4)\ \ (i = 0,1,2,3)$$

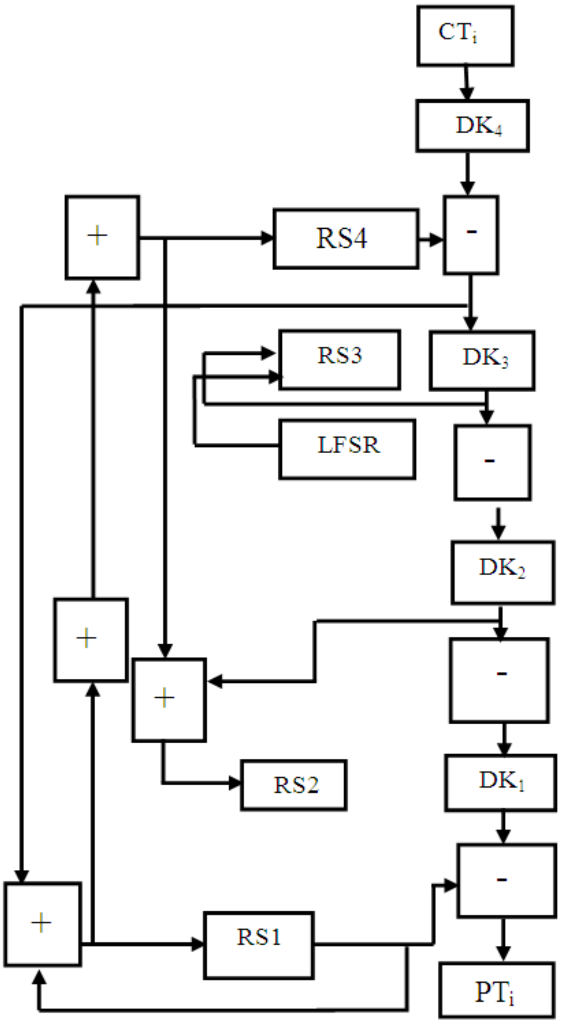

Fig. 3. Decryption for HB-2

2.2. NCL S-Box

The Null Conventional Logic (NCL) produce data only when both the inputs are present, if any one of the input is Null which is N then the output will be N.

The following **Table 1** shows the computation of NCL logic. It is clear that, from $S_0 \ldots S_3$ for all possible combination the output is T (i.e., True) else if it find Null logic (N) the output is N.

Apart from this, another input is utilized called as Intermediate (I) which is shown in **Table 2** in which it will get in to the Null condition only when both the inputs are null if any one input is in true condition (i.e., presence of data), the Intermediate I will be the output since the other input may get the data at a short interval of time.

Table 1. Null conventional logic

	S_0	S_1	S_2	S_3	N
S_0	T	T	T	T	F/N
S_1	T	T	T	T	F/N
S_2	T	T	T	T	F/N
S_3	T	T	T	T	F/N
N	F/N	F/N	F/N	F/N	F/N

Table 2. NCL based on Intermediate (I)

	S_0	S_1	S_2	S_3	I	N
S_0	T	T	T	T	I	I
S_1	T	T	T	T	I	I
S_2	T	T	T	T	I	I
S_3	T	T	T	T	I	I
I	I	I	I	I	I	I
N	I	I	I	I	I	N

Table 3. Hexadecimal notation of S-Boxes

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_1(x)$	8	6	5	F	1	C	A	9	E	B	2	4	7	0	D	3
$S_2(x)$	0	7	E	1	5	B	8	2	3	A	D	6	F	C	4	9
$S_3(x)$	2	E	F	5	C	1	9	A	B	4	6	8	0	7	3	D
$S_4(x)$	0	7	3	4	C	1	A	F	D	E	6	B	2	8	9	5

Based on the security criteria the Humming Bird algorithm will chose S-Box from four S-Boxes based on the **Table 3** for implementing the compact version of Humming Bird. The S-Box design consideration based on NCL-logic is expressed in following Equation 1 to 4:

$$\begin{aligned} S_1^{(0)}(x) ={} & x_3.\, x_2'\, .\, x_1'\, .x_0 + x_3'.\, x_2'\, .x_1 + x_3.\, x_2 \\ & .\, x_1.x_0 + x_2.x_1'\, .x_0' + x_3.\, x_2.\, x_1.x_0' + x_3'.\, x_1.\, x_0 \end{aligned} \tag{1}$$

$$\begin{aligned} S_1^{(1)}(x) ={} & x_3.\, x_2'\, .\, x_1'\, .x_0 + x_3.\, x_2.\, x_1.x_0 + x_3'.\, x_2. \\ & x_1.x_0' + x_3.\, x_2'\, .x_0' + x_3'.\, x_2'\, .x_0 + x_3.\, x_1'\, .\, x_0' \end{aligned} \tag{2}$$

$$\begin{aligned} S_1^{(2)}(x) ={} & x_3'.\, x_2'\, .x_1 + x_2'.\, x_1.\, x_0 + x_3.\, x_2.\, x_1. \\ & x_0' + x_3'.\, x_2'\, .\, x_0 + x_3.\, x_1'\, .\, x_0' + x_3'.\, x_2.\, x_1'\, .x_0 \end{aligned} \tag{3}$$

$$\begin{aligned} S_1^{(3)}(x) ={} & x_3.\, x_2'\, .\, x_1'\, .x_0 + x_3'.\, x_2.\, x_1.x_0\text{‘} + x_2' \\ & .x_1'.x_0' + x_3.\, x_2.\, x_1.x_0' + x_3'.\, x_1.x_0 + x_3'.\, x_2.\, x_1'\, .x_0 \end{aligned} \tag{4}$$

The Humming Bird algorithm consists of modulo operation based on key size. The operation varies from addition to subtraction based on encryption and decryption process. HB-2 consists of many rounds, based on linear and non linear operation**.** The S-Box is selected based on the above said criteria, since S-Box is the reputed components in the design implementation of Humming Bird algorithm. The improper design of S-Box leads to increasing power consumption and they are the vulnerable components to SCA Equation 5 to 12:

$$\begin{aligned} S_2^{(0)}(x) ={} & x_3'.\, x_2\, .x_1'.x_0 + x_3.\, x_2'\, .\, x_1.x_0' + x_3. \\ & x_2.\, x_1.x_0 + x_3'.\, x_2'.x_0 + x_2.\, x_1'\, .\, x_0' + x_3.\, x_1'\, .\, x_0' \end{aligned} \tag{5}$$

$$\begin{aligned} S_2^{(1)}(x) ={} & x_3'.\, x_2'\, .\, x_1.x_0\text{‘} + x_3'.\, x_2'.\, x_1'\, .x_0 + x_3' \\ & .\, x_2.x_0 + x_3.\, x_2'.\, x_1'.x_0 + x_3.\, x_2'.\, x_1.x_0 + x_3.\, x_1'\, .\, x_0' \end{aligned} \tag{6}$$

$$\begin{aligned} S_2^{(2)}(x) ={} & x_3'.\, x_2'\, .\, x_1.x_0\ \text{‘} + x_3'.\, x_2'.\, x_1'\, .x_0 + x_3 \\ & .\, x_2'.\, x_1.\, x_0 + x_3.\, x_1.x_0' + x_2.\, x_1'.\, x_0' + x_3.\, x_2.x_1' \end{aligned} \tag{7}$$

$$\begin{aligned} S_2^{(3)}(x) ={} & x_3'.\, x_2.\, x_1'\, .x_0 + x_3.\, x_2'.\, x_1.x_0' + x_3.\, x_2. \\ & x_1.x_0 + x_3'.\, x_1.\, x_0' + x_3.\, x_2'.\, x_1'\, .x_0 + x_3.\, x_2.x_1' \end{aligned} \tag{8}$$

$$\begin{aligned} S_3^{(0)}(x) ={} & x_3.\, x_2.\, x_1.x_0\text{‘} + x_2.\, x_1'.\, x_0 + x_3. \\ & x_2'.\, x_1'.x_0' + x_3'.\, x_1.x_0' + x_3.\, x_2.\, x_0 + x_3'.\, x_2'\, .\, x_1 \end{aligned} \tag{9}$$

$$\begin{aligned} S_3^{(1)}(x) ={} & x_3.\, x_2.\, x_1'.x_0 + x_3.\, x_2.\, x_1.x_0\text{‘} + x_3'.\, x_2. \\ & x_1.x_0 + x_3'.\, x_2'.\, x_1' + x_3.\, x_2'.\, x_1'\, .x_0' + x_2'.\, x_1\, .x_0' \end{aligned} \tag{10}$$

$$\begin{aligned} S_3^{(2)}(x) ={} & x_3'.\, x_2.\, x_1'.x_0' + x_2'.\, x_1'\, .x_0 \\ & + x_2'.\, x_1\, .x_0' + x_3.\, x_2\, .x_0 + x_3'.\, x_2'.\, x_1 \end{aligned} \tag{11}$$

$$\begin{aligned} S_3^{(3)}(x) ={} & x_3'.\, x_2'.\, x_1'.x_0 + x_3'.\, x_2.\, x_1'\, .x_0 + x_3'. \\ & x_2.\, x_1.x_0 + x_3.\, x_1.x_0 + x_3.\, x_2'.\, x_1'\, .x_0\text{‘} + x_3'.\, x_1.\, x_0' \end{aligned} \tag{12}$$

The proposed NCL S-Box has been developed using verilog HDL. The output for NCL S-Box is obtain only when the Data-0 and Data-1 is true on the other hand it shows the intermediate value as output if it experience any one null condition (i.e., Data-0 = T and Data-1 = N, output = I (Intermediate) or vice versa) Equation 13 to 16:

$$\begin{aligned} S_4^{(0)}(x) ={} & x_3.\, x_2'.\, x_1'.x_0' + x_2.\, x_1.x_0 + x_3'.\, x_2'. \\ & x_1.x_0' + x_3.\, x_2'.\, x_1.x_0 + x_3.\, x_2.\, x_1.x_0' + x_3'.\, x_1'\, .x_0 \end{aligned} \tag{13}$$

$$\begin{aligned} S_4^{(1)}(x) ={} & x_3'.\, x_2'.\, x_1.x_0' + x_3.\, x_2.\, x_1'\, .x_0' + x_3. \\ & x_2'.\, x_1.x_0 + x_2'.\, x_1'\, .x_0 + x_3'.\, x_2.x_1 + x_3.\, x_2'.\, x_1.x_0' \end{aligned} \tag{14}$$

$$\begin{aligned} S_4^{(2)}(x) ={} & x_3.\, x_2'.\, x_1'.x_0' + x_2.\, x_1.x_0 + x_3'\, .\, x_2. \\ & x_1'\, .x_0\text{‘} + x_2'\, .x_1'.\, x_0 + x_3.\, x_2'\, .x_1.x_0\text{‘} + x_3'.\, x_2'.\, x_0 \end{aligned} \tag{15}$$

$$\begin{aligned} S_4^{(3)}(x) ={} & x_3.\, x_2'.\, x_1'.x_0' + x_3'.\, x_2.\, x_1\text{‘}.x_0\text{‘} + x_3. \\ & x_1'\, .x_0 + x_3.\, x_2'\, .x_1.\, x_0 + x_3'.\, x_2.x_1 + x_3.\, x_2.\, x_1.x_0' \end{aligned} \tag{16}$$

Table 4. Evaluation for NCL S-Box

	Simulation Result		
		Output	
Mode	Input	S-Box	NCL S-Box
Encryption	9	11111110	1010101010101001
	26	01011110	0110011010100110
	106	11111101	1010101010100110
	122	00100101	0110010110010110
	158	11110100	1010101001100101
Decryption	32	10101011	1001100110011010
	51	10011001	1001011010010110
	156	11100011	1010100101011010
	185	00100100	0101100101100101
	203	10100110	1001100101101001

In second circumstance the output will be null if both the inputs are null (i.e., Data-0 = N and Data-1 = N, output = N (Null) or vice versa). Considering the time we are assuming 10 inputs (Encryption input = 5 and Decryption input = 5) both based on conventional S-Box and NCL S-Box as in **Table 4**.

For example the S-Box results in 11 11 11 01 for 106 where as the NCL S-Box 10 10 10 10 10 10 01 10 which will look like dual rail output. In account of decryption state the S-Box results in 11 10 00 11 for 156 which will be 10 10 10 01 01 01 10 10 in case of NCL S-Box. Similarly the decrypted output obtain as 01 01 10 01 01 10 01 01 and 10 01 10 01 01 10 10 01 for 185 and 203 respectively.

3. EXPERIMENTAL RESULT

The efficiency of the proposed NCL S-Box is determined in terms of area, Degree Of Confusion (DOC) and the frequency at which high throughput is obtained.

The **Table 5** shows that area consumed by the NCL S-Box in the targeted hardware is less when compared with the conventional one. In particular the logic element occupied by $S_4(x)$ is 149 when compared with rest of S-Box; this will lead to reduction in power consumption. The NCL S-Box produce a 16-bit encrypted output resulting in a high confusion rate, whereas we obtain 8- bit in case of conventional S-Box which made them more vulnerable to SCA. From **Table 6** we conclude that developed NCL S-Box runs at a frequency of 63.73 MHz which will contribute to higher throughput.

Table 5. Implementation of NCL S-Box

	S-Box			NCL S-Box		
		Size			Size	
TYPES	LE	Key	Block	LE	Key	Block
$S_1(x)$	163	128	64	158	128	64
$S_2(x)$	157	128	64	154	128	64
$S_3(x)$	159	128	64	153	128	64
$S_4(x)$	154	128	64	149	128	64

Table 6. Estimation of Fmax and DOC

Parameter	Maximum frequency (F_{Max}) (MHz)	Degree of Confusion (DOC)
S-Box	36.05	49.34
NCL S-Box	63.73	52.67

4. CONCLUSION

The most vulnerable component towards the Side Channel Attack (SCA) is S-Box. This study presents an optimized S-Box design for Humming Bird-2 which is an ultra lightweight cryptographic algorithm. The proposed S-Box is designed based on Null Conventional Logic (NCL) and developed using Verilog HDL. The implemented result was analyzed in terms of Logic Elements occupied (area), confusion rate which will determine the security level of the algorithm, finally the maximum frequency achieved in targeted hardware is computed. As of now, the paper contribute to cryptographic algorithm against the Side Channel Attack (SCA) in software level, in future this study would be extended on the hardware platform (i.e., physical attack) which is intrusion based on electromagnetic radiation, sound, heat etc. The Virtual Secure Circuit (VSC) can be taken in to account for hardware analyses of Side Channel Attacks.

5. REFERENCE

Abdelkrim, H., S.B. Othman, A.K.B. Salem and S.B. Saoud, 2012. Dynamic partial reconfiguration contribution on system on programmable chip architecture for motor drive implementation. Am. J. Eng. Applied Sci., 5: 15-24. DOI: 10.3844/ajeassp.2012.15.24

Aris, S., A. Messai, M. Benslama, M. Nadjim and M.M. Elharti, 2011. Integration of quantum cryptography through satellite networks transmission. Am. J. Applied Sci., 8: 71-76. DOI: 10.3844/ajassp.2011.71.76

Faudzi, A.A.M. and K. Suzumori, 2010. Programmable system on chip distributed communication and control approach for human adaptive mechanical system. J. Comput. Sci., 6: 852-861. DOI: 10.3844/jcssp.2010.852.861

Rajakumar, M.P. and V. Shanthi, 2014. Security breach in trading system-countermeasure using IPTraceback. Am. J. Applied Sci., 11: 492-498. DOI: 10.3844/ajassp.2014.492.498

Raphael, A.J. and V. Sundaram, 2012. New approaches to ancient crypto-steganography methods. Am. J. Applied Sci., 9: 40-46. DOI: 10.3844/ajassp.2012.40.46

Rohini, G. and S. Salivahanan, 2010. Optimal test time for system-on-chip designs using fuzzy logic and process algebra. J. Comput. Sci 6: 12-17. DOI: 10.3844/jcssp.2010.12.17

Salem, A.K.B., S.B. Othman and S.B. Saoud, 2010. Field programmable gate array-based system-on-chip for real-time power process control. Am. J. Applied Sci., 7: 127-139. DOI: 10.3844/ajassp.2010.127.139

Shivkumar, S. and G. Umamaheswari, 2014. Certificate authority schemes using elliptic curve cryptography, RSA and their variants-simulation using ns2. Am. J. Applied Sci., 11: 171-179. DOI: 10.3844/ajassp.2014.171.179

Takemura, T., 2010. A quantitative study on japanese workers' awareness to information security using the data collected by web-based survey. Am. J. Econ. Bus. Admin., 2: 20-26. DOI: 10.3844/ajebasp.2010.20.26