International Journal of Computer and Electrical Engineering, Vol. 2, No. 4, August, 2010
1793-8163

A Low-Power and High Throughput Reconfigurable Data Integrity Unit for Software Radio

L.Thulasimani and M.Madheswaran

***Abstract*—Software Radio terminals being able to reconfigure to various radio access systems, will require robust security mechanism to protect the mobile terminals from unauthorized access and to ensure the integrity of internal process involved in reconfiguration when exchanging information with the network. This paper aims to propose an energy efficient reconfigurable hardware architecture which supports data integrity for the secured downloading of software in reconfigurable receivers. Improved SHA-1(SHA-192) algorithm and a reconfigurable hardware design to support different protocols, which uses MD5/SHA 192 algorithm for integrity unit, is proposed. The hash algorithms MD-5, SHA-1, and the unified architecture of MD-5and SHA-192 have been implemented using Verilog, and their hardware utilization on the FPGA device is being compared**

***Index Terms*—SDR, Reconfigurability, SHA-192, Unified architecture, Hardware utilization.**

I. Introduction

Future multi function and multi mode mobile terminals based on SDR technologies will consist of individual dynamically configurable components that may use different access technologies(i.e. wireless LAN, cellular or satellite air interfaces or Bluetooth connection) to download configuration and radio software components that are necessary to reconfigure. Reconfigurability of radio terminals intrinsically provides the possibility of unintentionally but also deliberately corrupting the radio environment. To prevent such situations, reconfigurable systems will require security policies and architecture to cope with number of threats including unauthorized access, modification and to integrity.

Hardware architecture for high performance AES algorithm which is useful for SDR terminals has been implemented for encryption process [1]

Also radio security module that offers a SDR security architecture that enables separate software and hardware certification is being developed[2].

Security decipherment is achieved using the characteristics of the Field Programmable Gate Array, which allows the system to be arranged in a variety of different layouts[3].Cryptographic components are also exchanged for secure download. It includes the possibility to change any of the cryptographic components employed [4].

In this paper, reconfigurable hardware data integrity architecture has been proposed with an aim to provide secure download in SDR terminals. The proposed architecture is novel one which consumes less power than existing individual MD5 , SHA-1 hardware algorithms architectures. The area utilization of proposed architecture is analyzed with other existing hardware implementation and found that it is also very less.

Manuscript received JANUARY 20 2010.

L.Thulasimani is with the Department of ECE, PSG college of Technology,Coimbatore,INDIA E-mail: lthulasi@gmail.com.

Dr.M.Madheswaran is with Center for Advanced Research, Muthayammal Engineering College, Rasipuram.madheswara.dr@gmail.com

II. MD-5 And SHA-1 Algorithm

A. The MD5 Algorithm

MD5 was introduced in 1992 by Professor Ronald Rivest. It calculates a 128-bit digest for an arbitrary l-bit message. It is an enhanced version of its predecessor MD4[5][6].The algorithm could be described in two stages: Preprocessing and Hash Computation. Preprocessing involves padding a message, parsing the padded message into m-bit blocks, and setting initialization values to be used in hash computation. The final hash value generated by the hash computation is used to determine the message digest.

1) *Append Padding Bits* The b-bit message is padded so that a single 1 bit is appended to the end of the message, and then 0 bits are appended until the length of the message becomes congruent to 448, modulo 512.
2) *Append Length* A 64-bit representation of b is appended to the result of the padding. The resulting message has a length that is an exact multiple of 512 bits. This message is denoted here as Y.
3) *Initialize MD Buffer* Let A, B, C, D be 32-bit registers. These registers are initialized to the following values in hexadecimal, low-order bytes first:Word A: 01234567 Word B: 89abcdef Word C: fedcba98 Word D: 76543210
4) *Process Message in 16-Word Blocks* This is the heart of the algorithm, which includes four rounds of processing. The four rounds have similar structure but each uses different auxiliary functions F, G, H and I.

F (X , Y , Z) = (X and Y) or ((notx) and Y)
G(X, Y , Z) = (X and Z) or (Y and (notZ))
H (X , Y , Z) = X xor Y xor z

I (X , Y , Z) = Y xor (X or (notZ))

Each round consists of 16 steps and each step uses a 64-element table T [1 ... 64] constructed from the sine function. Let T[i] denote the i-th element of the table, which is equal to the integer part of 232 times abs(sin(i)), where i is in radians. Each round also takes as input the current 512-bit block Yq and the 128-bit chaining variable CVq. An array X of 32-bit words holds the current 512-bit Y.For the first round the words are used in their original order. The following permutations of the words are defined for rounds 2 through 4:

ρ2(i) = (1 + 5i) mod 16

ρ3(i) = (5+ 39) mod 16

ρ4(i) = 7i mod 16

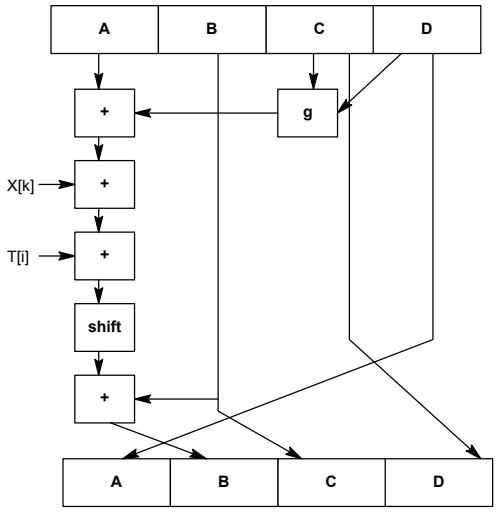

Figure 1. Operation of single step of MD5

The output of the fourth round is added to the input of the first round (CV,) to produce CVq+l.

5) *Output* After all L 512-bit blocks have been processed, the output from Lth stage is the 128-bit message digest. Figure 1 shows the operations involved in single step.The additions are modulo 232. Four different circular shift amounts S is used each round and are different from round to round. Each step is of the following form,

A -> D;B -> B + ((A + Funs (B , C , D) + X [K l + T [I]) < < s);C -> B;D -> C

III. The SHA-1 Algorithm

The Secure Hash Algorithm was developed by National Institute of Standards and Technology (NIST) and published as a federal information processing standard in 1993. It calculates a 160-bit digest for an arbitrary l-bit message. Preprocessing is done same as in MD5 except that an extra 32-bit register E is added with an initial value of C3D2E1F0. Other registers are assigned with higher order bytes first. For each block, it requires 4 rounds of 20 steps, resulting in a total of 80 steps, to generate the message digest[8].

A.Functions Used A sequence of logical functions f0, f1,..., f79 is used in the SHA-1. Each ft, 0 <= t <= 79, operates on three 32-bit words B, C, D and produces a 32-bit word as output. ft(B,C,D) is defined as follows, for words B, C, D,

ft(B,C,D) = (B and C) or ((not B) and D), for 0 <= t <= 19

ft(B,C,D) = B xor C xor D, for 20 <= t <= 39

ft(B,C,D) = (B and C) or (B and D) or (C and D), for 40 <= t <= 59

ft(B,C,D) = B xor C xor D, for 60 <= t <= 79

B.Constants Used A sequence of constant words K(0), K(1), ... , K(79) is used in the SHA-1. In hex these are given by

Kt = 5A827999 (0 <= t <= 19)

Kt = 6ED9EBA1 (20 <= t <= 39)

Kt = 8F1BBCDC (40 <= t <= 59)

Kt = CA62C1D6 (60 <= t <= 79)

C.Computing the Message Digest The message digest is computed using the final padded message. The computation uses two buffers, each consisting of five 32-bit words, and a sequence of eighty 32-bit words. The words of the first 5-word buffer are labeled A, B, C, D, E. The words of the second 5-word buffer are labeled H0, H1, H2, H3, H4. The words of the 80-word sequence are labeled W0, W1... W79. A single word buffer TEMP is also employed. To generate the message digest, the 16-word blocks M1, M2... Mn is processed in order. The processing of each Mi involves 80 steps. Single step operation of SHA-1 is shown in Fig.2. Before processing any blocks, the {Hi} are initialized as follows in hex:H0 = 67452301, H1 = EFCDAB89, H2 = 98BADCFE, H3 = 10325476, H4 = C3D2E1F0.

Now M1, M2... Mn is processed. To process Mi, we proceed as follows:

a. Divide Mi into 16 words W0, W1, ... , W15, where W0 is the left-most word.

b. For t = 16 to 79 let Wt = S1(Wt-3 XOR Wt-8 XOR Wt-14 XOR Wt-16).

c. Let A = H0,B = H1, C = H2, D = H3, E = H4.

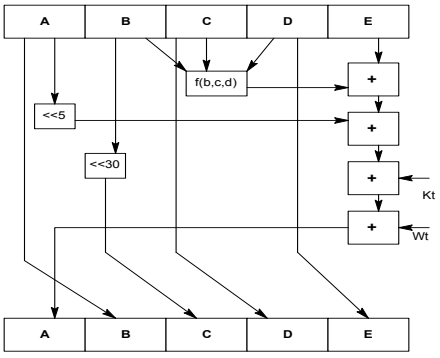

Figure 2. Operation of single step of SHA-1

d. For t = 0 to 79 do

TEMP = S5(A) + ft(B,C,D) + E + Wt + Kt;

E = D; D = C;C = S30(B);B = A;A = TEMP;

e. Let H0 = H0 + A,H1= H1 + B, H2 = H2+ C, H3 = H3 + D, H4 = H4 + E.

After processing Mn, the message digest is the 160-bit string represented by the 5 words H0 H1 H2 H3 and H4.

IV. Proposed SHA-192 Algorithm

The proposed SHA-192 is another improved version in SHA family. It may be used to hash message, M having a length of l bits, where 0<l<2^64. The algorithm uses, Six working variables of 32 bits each, A hash value of six 32-bit words. The final result of SHA-192 is the 192 bit message digest. The words of the message schedule are labeled W0, W1, W2…W79. The six working variables are labeled A,B,C,D,E and F.The words of the hash value are labeled H0(i),..,which will hold the initial hash value, and is replaced by each successive intermediate hash value(after each message block is processed)and ending with final hash value H(N).

A. SHA-192 preprocessing

The padding and appending of bits are done same as for MD5 and SHA-1. Before processing any blocks, the {Hi} are initialized as follows (in hex):

H0 = 67452301,H1 = EFCDAB89,H2 = 98BADCFE ,H3 = 10325476, H4 = C3D2E1F0,H5 = F9B2D834. The compression function of SHA-192 has been illustrated in Fig.3.

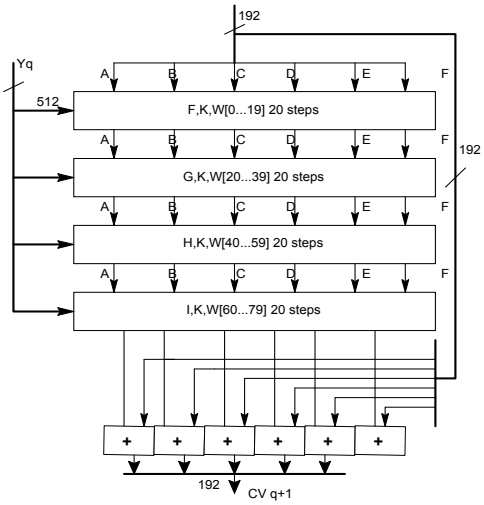

Figure 3. SHA-192 compression function

B. SHA-192 hash computation

A sequence of logical functions f0, f1,..., f79 is used in the SHA-192. Each ft, 0 <= t <= 79, operates on three 32-bit words B, C, D and produces a 32-bit word as output. ft(B,C,D) is defined as follows, for words B, C, D,

ft(B,C,D) = (B and C) or ((not B) and D), for 0 <= t <= 19

ft(B,C,D) = B xor C xor D, for 20 <= t <= 39

ft(B,C,D) = (B and C) or (B and D) or (C and D), for 40 <= t <= 59

ft(B,C,D) = B xor C xor D, for 60 <= t <= 79

A sequence of constant words K(0), K(1), ... , K(79) is used in the SHA-1. In hex these are given by Kt = 5A827999 (0 <= t <= 19)

Kt = 6ED9EBA1 (20 <= t <= 39)

Kt = 8F1BBCDC (40 <= t <= 59)

Kt = CA62C1D6 (60 <= t <= 79)

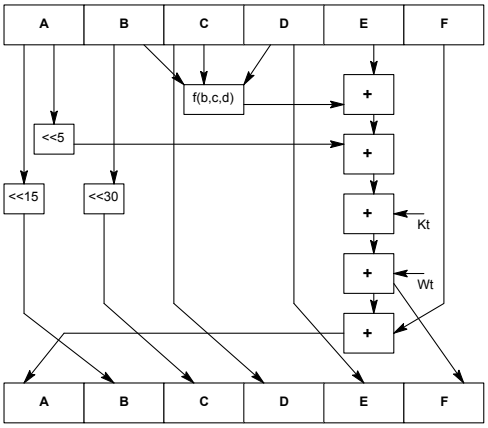

Figure 4. Operation of single step of SHA-192

Now M1, M2... Mn are processed. To process Mi, we proceed as follows:

a. Divide Mi into 16 words W0, W1... W15, where W0 is the left-most word.

b. For t= 0 to 15 Wt = Mi

For t = 16 to 79 let Wt = S1(Wt-3 XOR Wt-8 XOR Wt-14 XOR Wt-16).

c. Let A = H0, B = H1, C = H2,D = H3, E = H4,F = H5.

d. For t = 0 to 79 do

TEMP1 = S5(A) + ft(B,C,D) + E + Wt + Kt;

TEMP2 = S5(A) +A + ft(B,C,D) + E + Wt + Kt+F;

E = D;

D = C;

C = S30(B);

B = S15(A);

F = TEMP1;

A= TEMP2

e. Let H0 = H0 + A, H1= H1 + B,H2 = H2+ C,H3 = H3 + D, H4 = H4 + E,H5 = H5 + F.The single step operation of SHA-192 is shown in Fig.4.

After processing Mn, the message digest is the 192-bit string represented by the 6 words H0 H1 H2 H3 H4 and H5.

V. Unified Architecture Of Md-5 Sha-192

In the case of the MD5 operation the Data Transformation Unit uses only the four inputs/outputs B,C,D,E of each one of the four Data Transformation Rounds. The input/output named A,F is not used , for this hash function operation (MD5). This is due to the fact that MD5 processes on 128-bit blocks (4x32-bit) transformation blocks, instead of the 192-bit blocks that are used in SHA-192. The four Data Transformation Rounds are similar, but its one performs a different operation. MA components indicate modulo addition 232, while the shifters components define left shift rotations of the input data block[7].

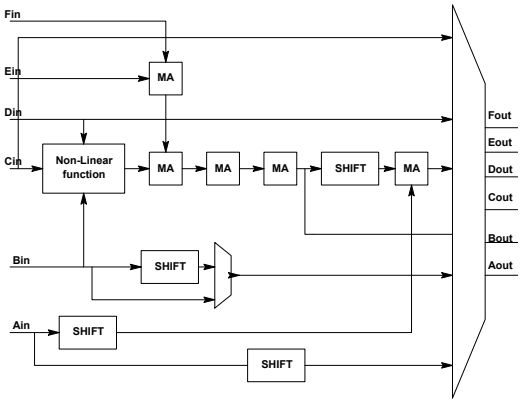

Figure 5. Data Transformation Of Unified Architecture

The Data Transformation Round I operation is based on a Nonlinear Function i transformation of the three of BIn, CIn, and DIn, inputs. Then, this result is added to the fourth input EIn with the input data block and the constant. That result is rotated to the right and therotated output data are added with the input DIn. There are four different nonlinear functions, one used for each Data Transformation Round, which perform the digital logic transformation according to equations in 2.1.

The Hash Function Core can be used alternatively for the operation SHA-192 hash function also. In this case, the Data Transformation Unit and the Data Transformation Rounds process the data in a different way, compared with MD5 operation mode, in order the Hash Function Core to perform efficiently as SHA-192.For the SHA-192 operation each Data Transformation Round operates on all the six 32-bit variables (inputs/outputs) and this is one of the basic differences compared with MD5 mode. Thus the combined architecture of MD-5 and SHA-192 results in reduced hardware utilization compared to the individual implementation of MD-5.In the hardware realization, a select line is inserted which selects the functionality of appropriate algorithm at each block. Data transformation for combined hash computation is as in Fig.*5*.

VI. Results and Discussion

The hardware architecture is implemented in Verilog, and synthesis is performed with Xilinx ISE 9.2i.Virtex II kit is chosen for downloading the synthesized code. The power analysis is done using Synopsys-Design vision. The hardware utilization are tabulated .

TABLE I Hardware Utilization of MD-5.

FPGA device : 2v4000bf957-6		
Allocated area	Used/Available	Utilization
I/Os	162/684	23%
Fun.Generators	724/46080	1%
CLB Slices	406/23040	1%
Dffs and Latches	298/46080	0%
frequency	57.36MHz	
Power consumption	4.55mW	
Throughput	458Mbps	

TABLE II. Hardware Utilization of SHA-1.

FPGA device : 2v4000bf957-6		
Allocated area	Used/Available	Utilization
I/Os	194/684	28%
Fun.Generators	2349/46080	5%
CLB Slices	1333/23040	5%
Dffs and Latches	1257/46080	2%
frequency	83.801 MHz	
Power consumption	15.49 mW	
Throughput	536Mbps	

Table 1 and 2 shows the hardware utilization summary of MD-5 and SHA-1 individually. Hence implementing these algorithms separately results in increased area occupation when compared with results in table 3.

TABLE III Hardware Utilization of Unified Architecture.

FPGA device : 2v4000bf957-6		
Allocated Area	Used/Available	Utilization
I/Os	195/684	28%
Fun.Generators	1275/46080	2%
CLB Slices	757/23040	3%
Dffs and Latches	1033/46080	2%
Throughput	845Mbps (MD5)	676Mbps (SHA192)
frequency	105.67MHz	
Power consumption	7.092mW	

From the above tabulation, it could be inferred that the device utilization is less in unified architecture compared with the individual implementation of MD-5 and SHA-1. The unified architecture of MD-5 and SHA-192 proved to consume less power and also efficient in computing the hash.

The comparison has been made between individual architectures of MD5 and SHA-192 with the unified architecture which contains both MD5 ans SHA-192 in the same module. Unified architecture will have some functionalities similar for both the MD5 ans SHA-192 algorithm so that the area utilization can be reduced and power consumption will also reduced.

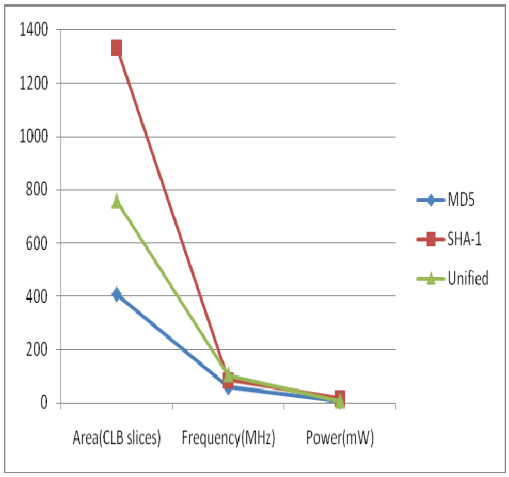

Figure 6.Area, frequency, power comparison.

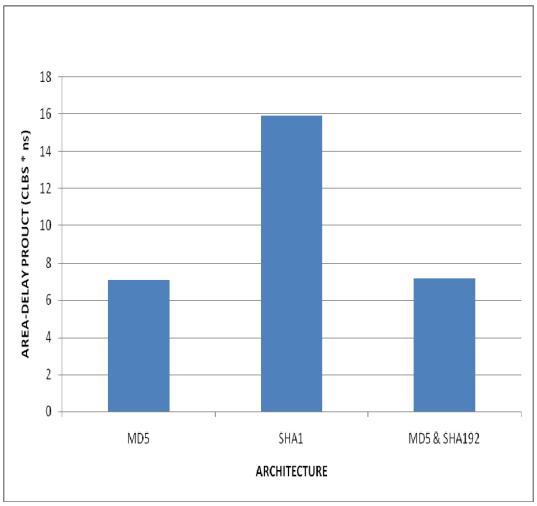

Figure 7.Area Delay product comparison.

The area, frequency and power consumption of MD5 and SHA-192 algorithm implemented in separate device and unified architecture is shown in figure 6. Analysis gives the area occupied ,frequency and power consumption when both the algorithms implemented in same architectures. The area delay product comparison of individual architectures and the unified architecture is given in figure 7. From which it could be inferred that the proposed combined architecture has better results.

The throughput analysis of individual and unified architectures is given in figure 8. Throughput is calculated using formulae,

$$Throughput = \frac{block\ size \times clock\ frequency}{latency}$$

Throughput of the unified architecture is also comparatively high.

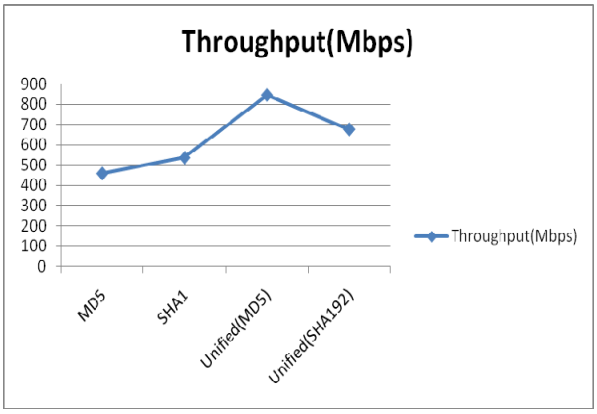

Figure 8.Throughput comparison.

The throughput of unified architecture achieves around 845 mbps for MD5 algorithm and 676mbps for proposed SHA-192 algorithm that is much better than individual implementations. the

Reference	*CLB Slices*	*Frequency in MHz*	*throughput*
[9]	14911/2304 0	43.47	171.2(MD5) 137.4(SHA-1)
Proposed Design	**757/23040**	**105.67**	**845(MD5) 676(SHA-192)**

The proposed design is compared with existing hardware implementation [9] and it found that the proposed architectures is better in all means that is, area utilization , frequency and throughput all very good than the results presented in [9]. And also power analysis is made for proposed architecture and shows it consumes only 7.09 mW which is not made in the previous work.

VII. Conclusion

Thus in this work, a VLSI architecture of the integrity unit for the reconfigurable receiver is presented. The architecture is reconfigurable in the sense that, it operates either to give MD-5 hash or the SHA-192 message digest. It guarantees security level in reconfigurable receivers requiring data integrity. The comparisons of synthesis results proved that the proposed integrity unit is better, compared to the individual implementation of the hash algorithms. The power consumption is also proved to be applicable for the reconfigurable receiver terminals

References

[1] Badillo,ClaudiaFeregrino-Uribe,Rene Cumplido,Morales-Sandoval.”FPGA Implementation and Performance evaluation of AES-CCM Cores for Wireless Networks.” In Proceedings of 2008 International Conference on Reconfigurable Computing and FPGaS

[2] Lozano, Marianela Garcia Moradi, Farshad Ayani, Rassul “ SDR: A Semantic Based Distributed Repository for Simulation Models and Resources” In Proceedings of Modelling & Simulation, 2007. AMS '07. First Asia international Conference on March 2007.

[3] Hironori uchikawa,Kenta umebayashi,Ryuji kohno "Secure download system based on software define radio composed of FPGAs" IEEE proceedings 2002.
[4] Chih Fung lam, Kei sakaguchi, Jun-ichi takada, kiyomichi araki"Radio Security Module that Enables Global Roaming of SDR Terminal while Complying with Local Radio Regulation" IEEE proceedings 2003.
[5] S. Dominikus, "A Hardware Implementation of MD4-Family Hash Algorithms",proceedings of IEEE International Conference on Electronics Circuits and Systems (ICECS'02), Vol. III,pp.1143-1146, Croatia, September 15-18, 2002.
[6] J. Deepakumara, H. M. Heys, and R. Venkatesan." FPGA implementation of MD5 hash algorithm." In Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2001),Toronto, Ontario, May 2001.
[7] N.sklavos,P P. kitsos K. Papadopoulos O. koufopavlou "Design, Architecture and Performance Evaluation of the Wireless Transport Layer Security" The Journal of Supercomputing, 36, 33–50, 2006 C_ 2006 Springer Science + Business Media, Inc. Manufactured in The Netherlands.
[8] Handbook of Applied Cryptography, by A. Menezes, P. van Oorschot, and S.Vanstone, Press, 1996.
[9] Esam Khan, Watheq El-Kharas hi M., and Mostafa Abd-El-Barr(2007), 'Design and Performance Analysis of A Unified Reconfigurable HMAC Hash Unit',IEEE Transaction on Circuits and Systems, Vol.54.N0.12 Pp.2683-2695.

L.Thulasimani has obtained her BE and ME degree from Coimbatore Institute of Technology, India in 1998 and 2001 respectively. She has started her teaching profession in the year 2001 in PSNA engineering college, Dindigul. At present she is an Lecturer in department of Electronic and Communication Engineering in PSG college of Technology, Coimbatore .She has published 4 research papers in International and National conferences. She is a part time research scalar in Anna University Chennai. Her areas of interest are Wireless security, Networking and signal processing. She is a life member of ISTE.

Dr. M. Madheswaran has obtained his Ph.D. degree in Electronics Engineering from Institute of Technology, Banaras Hindu University, Varanasi in 1999 and M.E degree in Microwave Engineering from Birla Institute of Technology, Ranchi, India. He has started his teaching profession in the year 1991 to serve his parent Institution Mohd. Sathak Engineering College, Kilakarai where he obtained his Bachelor Degree in ECE. He has served KSR college of Technology from 1999 to 2001 and PSNA College of Engineering and Technology, Dindigul from 2001 to 2006. He has been awarded Young Scientist Fellowship by the Tamil Nadu State Council for Science and Technology and Senior Research Fellowship by Council for Scientific and Industrial Research, New Delhi in the year 1994 and 1996 respectively. He has published 120 research papers in International and National Journals as well as conferences. His field of interest includes semiconductor devices, microwave electronics, optoelectronics and signal processing. He is a Senior member of IEEE, Fellow of IETE, and IE and member of ISTE