

ENHANCING DATA SECURITY USING DES WITH HARDWARE IMPLEMENTATION

¹LAKSHMI KANTHAM, ²Dr.S.RAVI

¹Research Scholar, Department of Electronics & Communication Engineering,
Sathyabama University, Tamil Nadu, India

²Professor, Department of Electronics & Communication Engineering,
Dr. M.G.R. University, Tamil Nadu, India

E-mail: ¹lks.sridhar@gmail.com, ²ravi_mls@yahoo.com

ABSTRACT

Network security has become an important issue in organization, business and public data handling and has grown exponentially in recent times. The threats to the valuable information on the network are becoming more widespread and more sophisticated. This work focuses on secure communication between end-points terminal and protect information during transmission. In this paper, DES algorithm is implemented on ARM processor using embedded C and provides better efficiency with hardware implementation is demonstrated.

Keywords: *Trusted Nodes, Data Security, Data Encryption Standard (DES)*

1. INTRODUCTION

Computer networks allow users to share and exchange a variety of information. Information on the network is subject to vulnerabilities, such as unauthorized access to data, clandestine alteration of data, identity fraud, denial of service etc., Data Security is a challenging issue in today IT services. Therefore, it is necessary to protect information against unauthorized access or damage among interoperated nodes. In this paper Cryptography concept were used to protect data transmission over unreliable network. The OSI stack is responsible for end-to-end communication. This model enforce strict boundaries between layer i.e., data are kept within a given layer. The advantage of OSI layered architecture is to enable fast development between the interoperable node and improve communication protocols. However, the layered architecture limit the overall performance, due to lack of coordination among layer. Such limitation can be overcome by cross-layer protocol. The cross layer protocol[13] removes such strict boundaries and it allows coordination and interaction of protocols crossing different layers by utilizing the functionalities associated to the original layer. In [1], each node has individual Intrusion Detection Systems (IDS). This system will regulate the network structure and assist in monitoring the network traffic. Thus, each node in the network will be treated as trusted node, but there is possibility for entry of malicious node

and this node will disturb the entire network operation, like packet drop, delay, link break, message tampering, stealing information, fake routing etc., Hence to establish trustworthiness among the node, the dynamic network traffic is monitored and combined with network structure table (Historical hop table [1]) to identify the malicious node. There are several ways to provide security for the information. Here the combination of IDS and Cryptographic algorithm concept are used to provide authentication (prevent untrusted nodes) and perform secure data transfer between end points.

2. CROSS-LAYER ARCHITECTURE

Cross Layer Protocol is a suitable approach for secure and reliable data transfer among the node in the wide spread network[12]. In this paper top-to-down cross layer protocol is preferred when compare to the down-to-up approach, due to following metrics:

- (i) throughput
- (ii) packet size
- (iii) cost
- (iv) Input file size and
- (v) Delay between packets.

The following Table 1 shows the comparison between various cross layers design approaches[14].

Table: 1 Comparison Between Cross-Layer Design Approaches Based On The Direction Of Information.

Factors	Approaches		
	Downward	Upward	Hybrid
Cost	Minimizing cost for a given quality	Cost is specified by lower layers	Minimizing cost
Packet loss	Efficient	Less efficient	Efficient
Delay	Efficient	Less efficient	Efficient
Throughput	Efficient	Less efficient	Efficient
Complexity	Less complexity	Less complexity	Complexity

Fig:1 Shows the proposed cross-layer architecture with different functionalities at different layer. This structure protocol used to provide security (confidentiality, access control, integrity non-repudiation and authentication) for transmission information between the trusted nodes. In this paper, it is focused to design an Embedded Cryptographic algorithm (DES), developed with Keil C platform. The secure data transfer is achieved between end points by running embedded cryptographic algorithm (DES) in each trusted node (Refer Fig:2). This proposed trusted node will support top to down cross layer protocol (Refer Fig: 1) and deliver a system that will maximize the performance. The choice of preferring the Embedded Hardware [7] (ARM Processor) is simple, high integration, small size, easy to transplant, increased processing power, speed besides the cost[10] when compared to FPGA [2].

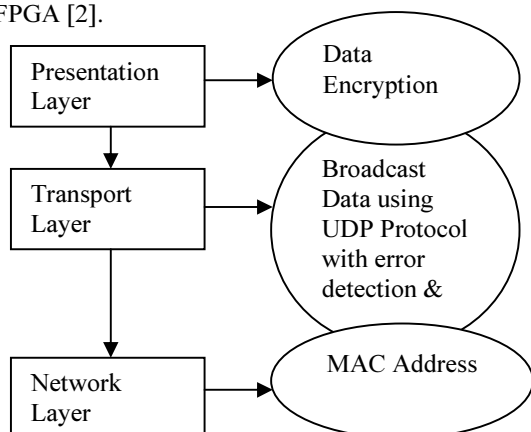


Fig:1 Different Layer With Their Functionalities

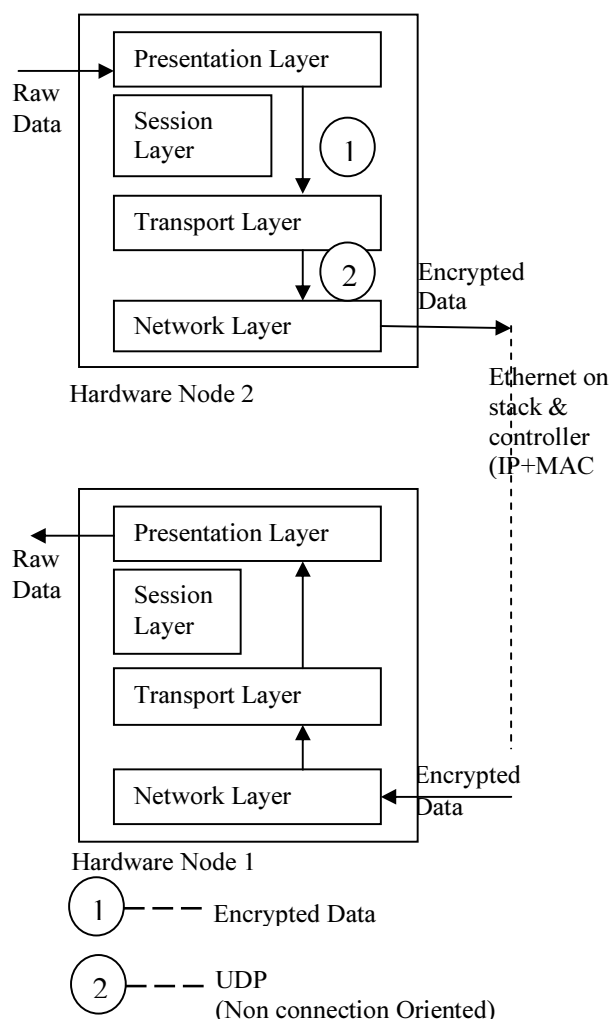


Fig : 2 Embedded Cryptography Algorithm Hardware With Keil C Platform

3. NEED FOR CRYPTOGRAPHY

Cryptography comes from Greek word means “Hidden Writing”. It converts readable data or cleartext into encoded data called ciphertext. It is an important technique needed in today’s wired and wireless network, in order to prevent sensitive data from being hacked by an attacker. Since, wired and wireless networks are more vulnerable to sniffing. There are two basic encryption methods[4] (i) Symmetric Encryption and (ii) Asymmetric Encryption.

3.1.1 Symmetric Encryption (Private-Key Cryptography)

Symmetric Encryption is an oldest and most secure encryption method and uses identical key to encrypt and decrypt the data. To provide privacy

the key should be kept secretly. It can be divided into stream ciphers and block ciphers. Stream ciphers encrypt the bits of the message one at a time, and block ciphers take a number of bits (64 bits) and encrypt them as a single unit or Block.

Some of the popularly well-known symmetric algorithms are: DES, Triple-DES, 3DES, IDEA, AES, Serpent, CAST5, RC4, BLOWFISH, TWOFISH etc.,. The encryption and decryption process in this method is generally faster, consume less computing power than with asymmetric encryption, but key distribution can be difficult.

3.1.2 Asymmetric Encryption (Public-Key Cryptography)

Asymmetric encryption method is potential and more secure than symmetric methods. It uses pairs of keys. One ("public key,") is used for encryption and the other one ("private" key") for decryption. Public key is freely available to all end user to encrypt the data before sending them and can be decrypted only by the authorized person who holds the private key. However the private key is kept secret. Thus, use of two keys overcomes a major weakness in symmetric key cryptography, since a single key does not need to be securely managed among multiple users. Some of the Well-known asymmetric algorithms are RSA, DSA, etc.,.

However, asymmetric algorithms are much slower than symmetric method. Therefore, a combination of both is being used in many of the applications.

3.1.3 Protocols Vulnerable to Sniffing

Protocols that are susceptible to sniffers include:

- Telnet & Rlogin: Keystrokes including user names and passwords
- HTTP: Data sent in Clear Text
- SMTP: Password and data sent in clear text
- NNTP: Passwords and data sent in clear text
- POP: Passwords and data sent in clear text
- FTP: Passwords and data sent in clear text
- IMAP: Passwords and data sent in clear text

The above protocols send a file over the network in clear text format and this will allow the attacker to eavesdrop the packet easily from the network.

3.1.4 Safeguard against Sniffing

Sniffing can be avoided in the network by using strong encryption and decryption technique to protect confidential information. SSH can use instead of Telnet, SCP (secure Copy) instead of FTP, SSL for SMTP etc., to protect wireless network users against sniffing attacks.

4.Data Encryption Standard (DES)

DES algorithm is used as an Embedded Cryptographic with Keil C Platform. The purpose of DES algorithm is to supplement the entire network trusted nodes. Each of the trusted nodes can provide their own security and monitor any anomaly activity in the network. Data Encryption Standard (DES)[3] is a widely used method of data encryption. This encryption technique was developed in the 1970's and was adopted by the U.S. Department of Defense. Its purpose is to provide a standard method for protecting sensitive commercial and unclassified data. Fig 3 and Appendix 1 shows the Basic structure of DES algorithm and internal structure of DES. DES[4] is a Feistel cipher which processes plain text blocks of $m=64$ bits, producing 64 bit ciphertext C blocks. The effective size of the secret key K is $K=56$ bits; more precisely, the input key K is specified as a 64 bit key, 8 bits of which (bits 8, 16, 24, 32, 40, 48, 56, and 64) can be used as parity bits. The 256 keys implement 2^{56} of the 2^{64} possible bijections on 64-bit blocks. Additionally, parity bits can be introduced to reduce the effective key size from 64 to 56 bits, and reduce the cost of exhaustive key search by a factor of 256. The same key is used to decrypt the ciphertext block.

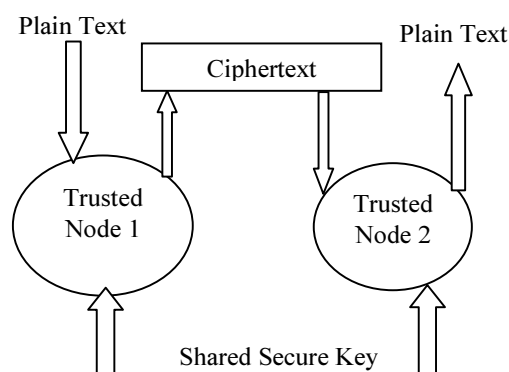


Fig:3 Basic Structure of Block Cipher Algorithm

4.1 DES Encryption and Decryption Algorithm

4.1.1 Encryption

INPUT : Plaintext $m_1 \dots m_{64}$; 64-bit, key $K = k_1 \dots k_{56}$

OUTPUT : 64-bit ciphertext block $C = c_1 \dots c_{64}$.

1. **Key Schedule:** Compute 16- 48-bit round keys K_i ($1 \leq i \leq 16$) from Key generation

Algorithm.

2. $(L_0, R_0) \leftarrow IP(m_1 m_2 \dots m_{64})$. (Use Initial Permutation (IP) [4] to permute bits)

3. 16 round: for i from 1 to 16, compute L_i and R_i using Equations

$$L_i = R_{i-1},$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i), \text{ where}$$

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i)).$$

4. Swap 2 half blocks out from round 1; $b_1 b_2 \dots b_{64} \leftarrow (R_{16}, L_{16})$.

5. Apply Inverse Initial Permutation IP^{-1} [4]

$C \leftarrow IP^{-1}(b_1 b_2 \dots b_{64})$. (Transpose using IP^{-1} Table [4]; $C = b_{40} b_8 \dots b_{25}$)

4.1.2 Key Generation

In the Input Key of 64 bits, each 8th bit (i.e., 8, 16, 24, 32, 40, 48, 56, 64) are unused. The 56-bit key is first subjected to permutation Permuted Choice 1 (PC-1). The resulting 56-bit key is then treated as 2- 28-bit quantities, labeled C_{i-1} and D_{i-1} . At each round, C_{i-1} and D_{i-1} are separately subjected to a circular left shift of 1 or 2 bits. These shifted values serve as input to the next round. They also serve as input to Permuted Choice 2(PC-2), which produce a 48-bit output that serves as input to the function $f(R_{i-1}, k_i)$.

4.1.3 Decryption

INPUT: 64-bit ciphertext block $C = c_1 \dots c_{64}$ & key $K = k_1 \dots k_{56}$.

OUTPUT: plaintext $m_1 \dots m_{64}$; 64-bit

1. **Key Schedule:** Compute 16-48-bit round keys K in reverse order i.e., from $K_{16}, K_{15} \dots K_1$ using key generation Algorithm.

2. $(L_0', R_0') = (R_{16}, L_{16}) \leftarrow IP(c_1 c_2 \dots c_{64})$. (Use IP Table[4] to permute bits; split the result into left and right 32-bit halves)

3. 16 round: for i from 1 to 16, compute L_i' and R_i' using Equations

$$L_1' = R_0' = L_{16}, R_1' = L_0' \oplus f(R_0', K_1')$$

$$= R_{16} \oplus f(L_{16}, K_1').$$

4. Swap 2 half blocks out from round 16

$$(R_{16}, L_{16}) \leftarrow (L_{16}, R_{16}).$$

5. Apply Inverse Initial Permutation IP^{-1} [4]

$C \leftarrow IP^{-1}(b_1 b_2 \dots b_{64})$. (Transpose using IP^{-1} Table [4])

5. DES IMPLEMENTATION IN HY-LPC1788 –CORE BOARD

Embedded Hardware with the combination of Cryptographic Algorithm (DES)[11] is convenient, flexible, high integrated & customizable feature[8]. Thus, DES algorithm is implemented in LPC1788. The HY-PLC1788 [5] is a Cortex-M3 microcontroller for embedded applications featuring a high level of integration and low power consumption[9] at frequencies of 120 MHz. Features include 512 kB of flash memory, 96 kB of data memory, Ethernet, USB 2.0 Host/OTG/Device, 8-channel DMA controller, 5 UARTs, 2 CAN channels, 3 SSP/SPI, 3 I2C, I2S, 8-channel 12-bit ADC, 10-bit DAC, QEI, LCD controller, SD/MMC and up to 165 general purpose I/O pins. This board is mainly used in network communications, automotive electronics, medical electronics, industrial control system, consumer electronics, and other aspects.

6. RESULTS AND DISCUSSION

In this paper, DES algorithm [6] is implemented on ARM LPC1788 Hardware Board using Embedded C. Two Hardware Board were used, -one of the board treated as Hardware Node-1(Source) and other one is Hardware Node-2 (Destination) - Refer Fig 2. The communication between the Trusted (Hardware) Nodes for data transfer take place through encrypted manner through the top-to-down cross layer protocol approach. Here, one node act as source and it accepts information from the other node (destination). Each node is assigned with logical address to facilitate the routing of packets over the network, logical address (IP Address) for hardware node-1 is 192.168.000.100 (Source) and other node (hardware node-2) is 192.168.000.99 (destination). The data in the individual node transfer to other node through MAC layer concept i.e., either Physical (or) Data link layer. Data link Layer or Physical Layer consists of the 32-bit code MAC header info and authenticating is updated at the end of frame and each node are assigned 6-byte MAC address(Refer Table 3). Therefore the total size of the data structure is maintained in the network layer is $6+6+1+N+4$ bytes (Refer Table 2 and 3). The 32-bit IP addresses are used, the address resolution protocol (ARP) is intrinsically built so that two stations do not have address conflict.

Node id	Station MAC	Entered id
Station node	=SRCMAC	0xA1, 0xA2, 0xA3, 0xA4, 0xA5, 0xA6 (Node 1)
Other nodes	=DSTMAC	0x90, 0x2B, 0x34, 0xaAB, 0x09, 0x8C (Node 2)

Table :2 Data Structure in MAC layer for node 1

Source MAC	Destination MAC	Beacon	Raw Data	32-Bit Authenticating
(6 bytes)	(6 bytes)	(1 byte)	(N bytes)	(4 bytes)

Table :3 Data Structure in MAC layer for node 2

Destination MAC	Source MAC	Beacon	Raw Data	32-Bit Authenticating
(6 bytes)	(6 bytes)	(1 byte)	(N bytes)	(4 bytes)

Table: 4 Node identified by MAC address

The following figure shows the result of the DES implementation on ARM LPC1788. Here Hyper Terminal is used to transmit and receive the encrypted/decrypted data. Fig 5 & Fig 6 shows the secure communication between two trusted nodes and untrusted node with different key.

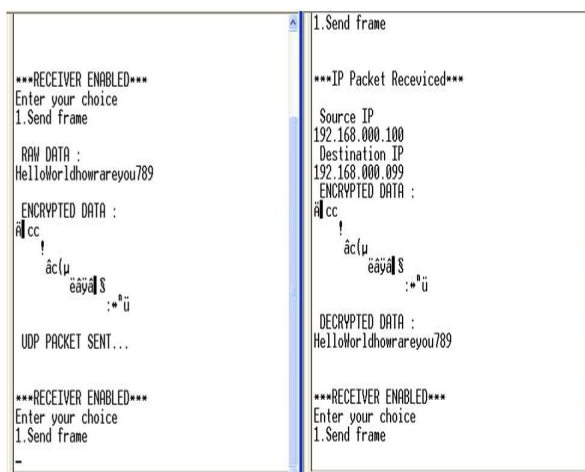


Fig: 5 Communication between Trusted Nodes.

Fig: 6 Untrusted Node with different key

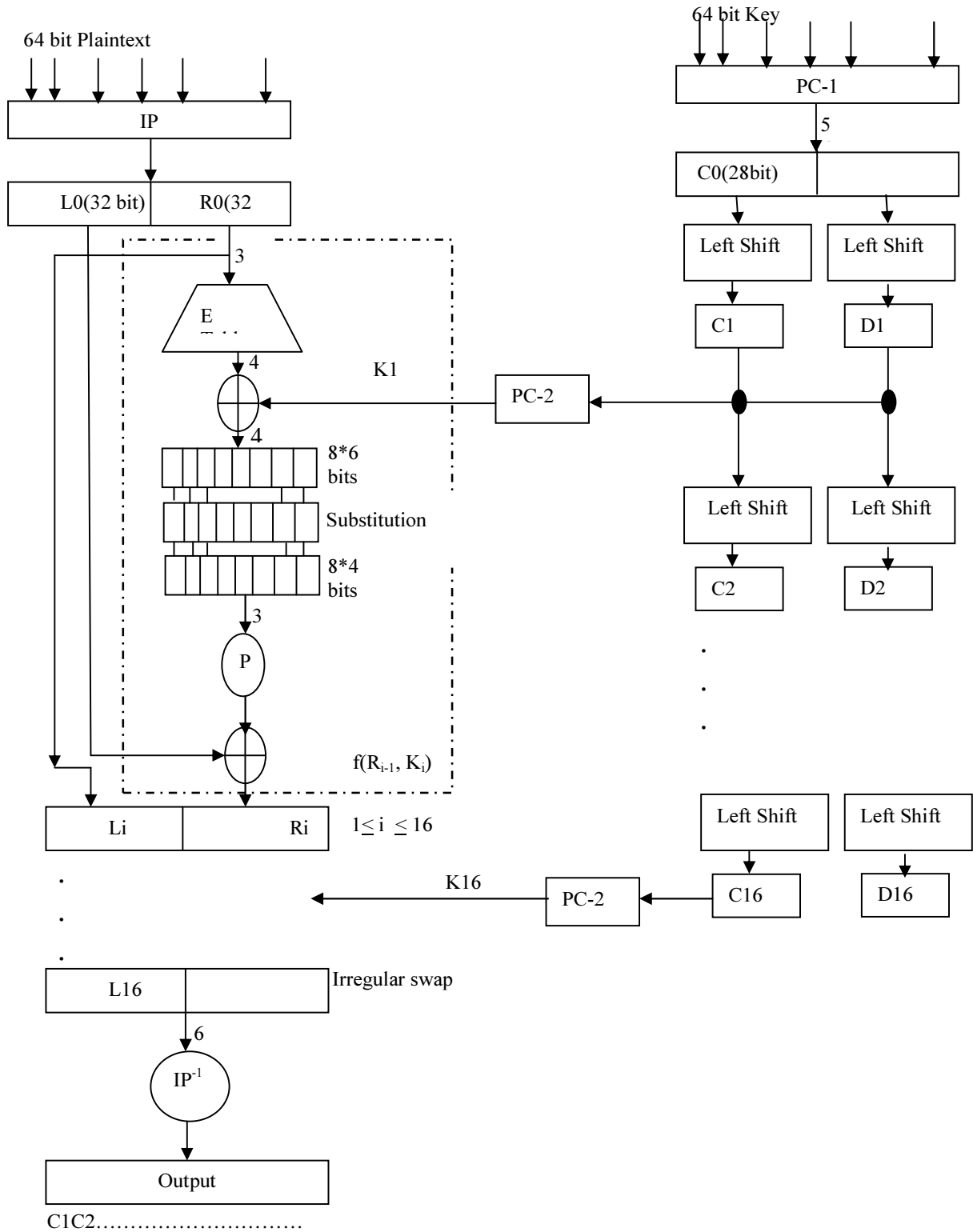
7. CONCLUSION

In this work, an encryption algorithm is embedded with the hardware core (ARM processor) to provide security for data transfer between multiple mobile nodes or machine to machine communication (trusted nodes) with makes a better transmission between the network nodes. It shows that encryption technique can act as supplement to the network for providing data security. Additionally, node can be protected in conjunction with security control (such as physical access, authentication, authorization or network controls) to adequately ensure the confidentiality, integrity and availability of the node link.

REFERENCES:

- [1] S.Lakshmi Kantham & Dr.S.Ravi, "SOC Based Self-Healing Architecture for Data Security", *International Journal of Computer Application (0975-8887)*, Vol. 64- No.21, February 2013, pp.11-16 .
- [2] T.Ravichandra Babu , K.V.V.S.Murthy & G.Sunil, "AES Algorithm Implementation using ARM Processor", *2nd International Conference and workshop on Emerging Trends in Technology (ICWET) 2011 Proceedings published by International Journal of Computer Applications® (IJCA)*, pp.24-59.
- [3] Amol D. Tupkar, Prof. U.A. Rane, "ARM Microcontroller Implementation of DES Using Concept with Time-Variable Key", *International Journal of advancement in electronics and computer engineering (IJAEECE)* Vol. 1, Issue 2, May 2012, pp.62-68.
- [4] Stallings, William, "Cryptography and Network Security", Prentice Hall, 1998.

- [5] UM10470 LPC178x/7x User manual
- [6] C Implementation of Cryptographic Algorithms, Texas Instrumentation.
- [7] Ho Won Kim, and Sunggu Lee, "Design and Implementation of a Private and Public Key Crypto Processor and Its Application to a Security System", *IEEE Transactions on Consumer Electronics*, Vol. 50, No. 1, February 2004 pp.215-224.
- [8] Ali E. Taki El_Deen, Noha A. Hikal, "Microcontroller Application in Cryptography Techniques", *Canadian Journal on Electrical and Electronics Engineering* Vol. 1, No. 4, pp:68-70.June 2010,
- [9] Yadollah Eslami, Ali Sheikholeslami, P. Glenn Gulak, Shoichi Masui, and Kenji Mukaida "An Area-Efficient Universal Cryptography Processor for Smart Cards", *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, Vol. 14, NO. 1, pp. 43-56, January 2006.
- [10] Helena Rif'a-Pous and Jordi Herrera-Joancomart, "Computational and Energy Costs of Cryptographic Algorithms on Handheld Devices" *Future Internet* 2011, 3, pp.31-48.
- [11] Karim Moussa Ali Abd El-Latif, Hesham Fathi Ali Hamed and El-Sayed Abd El-Hameed, "Hardware Implementation of DES Using Pipelining Concept with Time-Variable Key", *IEEE 22nd International Conference on Microelectronics (ICM 2010)*
- [12] Jaydip Sen, Shomik Bhattacharya, "A Survey on Cross-Layer Design Frameworks for Multimedia Applications over Wireless Networks" *International Journal of Computer Science and Information Technology (IJCSIT)*, Vol: 1, No:1, pp. 29-42, June 2008.
- [13] F. Aune, "Cross-Layer Design Tutorial," *Norwegian University of Science and Technology*, Dept. of Electronics and Telecommunications, Trondheim, Norway, 2004.
- [14] K. M. El Defrawy, M. S. El Zarki, and M. M. Khairy, "Proposal for a cross-layer coordination framework for next generation wireless systems," *ACM International Conference On Communications And Mobile Computing*, pp. 141 – 146, 2006.



Appendix :1 Internal Structure of DES Algorithm