IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

Optimized Encrypted Entries for S & Pp Box

Jambunathan S[1], Sathish A[2], Shankar R[3]

M.Phil Research Scholar, Department of Computer Science, Chikkanna Govt Arts College, Tiruppur[1]

Assistant Professor, Department of Computer Science, Maharaja Arts & Science College, Coimbatore[2]

Assistant Professor, Department of Computer Science, Chikkanna Government Arts College, Tiruppur[3]

Abstract: Advanced Encryption Standard is a reliable cryptosystem which is used in many applications. It consists of four rounds and byte substitution is the major of those rounds. Procedure of this round is to substitute the values with the entries of Substitution-Box which is known as S-Box. Its entries are pre-computed and stored on lookup tables (ROMs) to avoid tedious real time computations. There also have possible chance of attacks (known & Side-channel) proposed due to direct storage of S-Box and it spoils the overall security of the cryptosystem. To protect against such vulnerabilities, tracking on the security of the S-Box is necessary. Existing system that provides such security to the S-Box has time and space complexities in storage and searching mechanism. Our objective is to propose an optimized system from all the existing along with the space and time complexity of direct S-Box.

Keywords: S-BOX, IS-BOX, PP-BOX, IPP-BOX, S-Box, IS-Box

I. INTRODUCTION

Network security consists of the requirements and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose or are assigned an ID and password or other authenticating information that allows them access to information and programs within their authority.

Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. It does as its title explains: It secures the network, as well as protecting and overseeing operations being done. The most common and simple way of protecting a network resource is by assigning it a unique name and a corresponding password. There are several security domains followed in network security concepts. In our project we use Cryptography as our security domain and it's explained below.

II. CRYPTOGRAPHY

In general Cryptography means "the art of writing or solving codes". Cryptography is derived from the Greek words: kryptós, "hidden", and gráphein, "to write" or "hidden writing". People who study and develop cryptography are called cryptographers. The study of how to circumvent the use of cryptography for unintended recipients is called cryptanalysis, or code breaking.

Cryptography and cryptanalysis are sometimes grouped together under the umbrella term cryptology, encompassing the entire subject. In practice, "cryptography" is also often used to refer to the field as a whole, especially as an applied science. Cryptography is the practice and study of techniques for secure communication in the presence of third parties. Cryptography can generally defined as "The art of protecting information by transforming it (encrypting it) into an unreadable format, called cipher text. Only those who possess a secret key can decipher (or decrypt) the message into plain text. This cryptography concept can easily be understand by the following Fig.1.1.

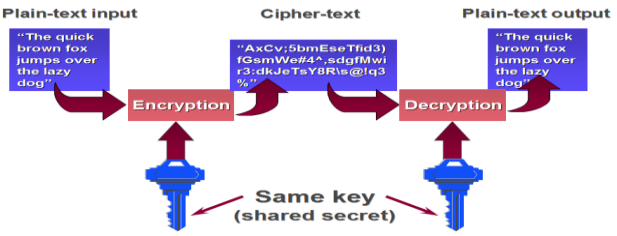

Fig.1.1. general block diagram for Cryptography

Cryptography technique is classified into two different types and they are Symmetric key Cryptography and Asymmetric key Cryptography.

2.1 SYMMETRIC KEY CRYPTOGRAPHY

An encryption system in which the sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message. Contrast this with public-key cryptology, which utilizes two keys - a public key to encrypt messages and a private key to decrypt them. Symmetric-key systems are simpler and faster, but their main drawback is that the two parties must

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

somehow exchange the key in a secure way. Public-key encryption avoids this problem because the public key can be distributed in a non-secure way, and the private key is never transmitted. Symmetric-key cryptography is sometimes called secret-key cryptography.

The most popular symmetric-key system is the Data Encryption Standard (DES) and Advanced Encryption Standard (AES).

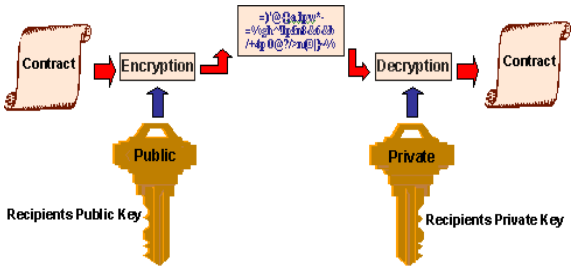

Fig 1.2 Symmetric key Cryptography

2.2 ASSYMETRIC KEY CRYPTOGRAPHY

Public key cryptography was invented in 1976 by Whitfield Diffie and Martin Hellman. For this reason, it is sometime called Diffie-Hellman encryption. It is also called asymmetric encryption because it uses two keys instead of one key (symmetric encryption).

A cryptographic system that uses two keys -- a public key known to everyone and a private or secret key known only to the recipient of the message. When John wants to send a secure message to Jane, he uses Jane's public key to encrypt the message. Jane then uses her private key to decrypt it. An important element to the public key system is that the public and private keys are related in such a way that only the public key can be used to encrypt messages and only the corresponding private key can be used to decrypt them. Moreover, it is virtually impossible to deduce the private key if you know the public key.

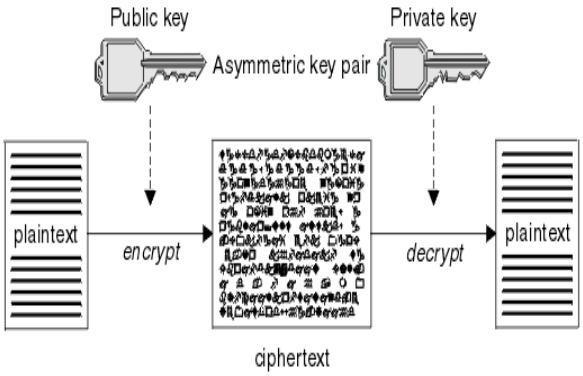

Fig 1.3 Asymmetric key Cryptography

III. KNOWN ATTACKS

For cryptographers, a cryptographic "break" is anything faster than a brute force—performing one trial decryption for each key (see Cryptanalysis). This includes results that are infeasible with current technology. The largest successful publicly known brute force attack against any block-cipher encryption was against a 64-bit RC5 key distributed.net in 2006.

AES has a fairly simple algebraic description. In 2002, a theoretical attack, termed the "XSL attack", was announced by Nicolas Courtois and Josef Pieprzyk, purporting to show a weakness in the AES algorithm due to its simple description. Since then, other papers have shown that the attack as originally presented is unworkable; see XSL attack on block ciphers.

During the AES process, developers of competing algorithms wrote of Rijndael, "...we are concerned about use...in security-critical applications." However, in October 2000 at the end of the AES selection process, Bruce Schneier, a developer of the competing algorithm Twofish, wrote that while he thought successful academic attacks on Rijndael would be developed someday, "I do not believe that anyone will ever discover an attack that will allow someone to read Rijndael traffic."

On July 1, 2009, Bruce Schneier blogged about a related-key attack on the 192-bit and 256-bit versions of AES, discovered by Alex Biryukov and Dmitry Khovratovich, which exploits AES's somewhat simple key schedule and has a complexity of 2^{119}. In December 2009 it was improved to $2^{99.5}$. This is a follow-up to an attack discovered earlier in 2009 by Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolić, with a complexity of 2^{96} for one out of every 2^{35} keys.

Another attack was blogged by Bruce Schneier on July 30, 2009 and released as a preprint on August 3, 2009. This new attack, by Alex Biryukov, Orr Dunkelman, Nathan Keller, Dmitry Khovratovich, and Adi Shamir, is against AES-256 that uses only two related keys and 2^{39} time to recover the complete 256-bit key of a 9-round version, or 2^{45} time for a 10-round version with a stronger type of related sub key attack, or 2^{70} time for an 11-round version. 256-bit AES uses 14 rounds, so these attacks aren't effective against full AES. In November 2009, the first known-key distinguishing attack against a reduced 8-round version of AES-128 was released as a preprint. This known-key distinguishing attack is an improvement of the rebound or the start-from-the-middle attacks for AES-like permutations, which view two consecutive rounds of permutation as the application of a so-called Super-Sbox. It works on the 8-round version of AES-128, with a time complexity of 2^{48}, and a memory complexity of 2^{32}.

In July 2010 Vincent Rijmen published an ironic paper on "chosen-key-relations-in-the-middle" attacks on AES-128. The first key-recovery attacks on full AES were due to Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger, and were published in 2011. The attack is based on bicliques and is faster than brute force by a factor of about four. It requires $2^{126.1}$ operations to recover an AES-128 key. For AES-192 and AES-256, $2^{189.7}$ and $2^{254.4}$ operations are needed, respectively.

3.1 SIDES-CHANNEL ATTACKS

Side-channel attacks do not attack the underlying cipher, and thus are not related to security in that context. They rather attack implementations of the cipher on systems

DOI 10.17148/IJARCCE.2016.51031

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

which inadvertently leak data. There are several such known attacks on certain implementations of AES. In April 2005, D.J. Bernstein announced a cache-timing attack that he used to break a custom server that used Open SSL's AES encryption. The attack required over 200 million chosen plaintexts. The custom server was designed to give out as much timing information as possible (the server reports back the number of machine cycles taken by the encryption operation); however, as Bernstein pointed out, "reducing the precision of the server's timestamps, or eliminating them from the server's responses, does not stop the attack: the client simply uses round-trip timings based on its local clock, and compensates for the increased noise by averaging over a larger number of samples." In October 2005, Dag Arne Osvik, Adi Shamir and Eran Tromer presented a paper demonstrating several cache-timing attacks against AES. One attack was able to obtain an entire AES key after only 800 operations triggering encryptions, in a total of 65 milliseconds. This attack requires the attacker to be able to run programs on the same system or platform that is performing AES.In December 2009 an attack on some hardware implementations was published that used differential fault analysis and allows recovery of a key with a complexity of 2^{32}.

In November 2010 Endre Bangerter, David Gullasch and Stephan Krenn published a paper which described a practical approach to a "near real time" recovery of secret keys from AES-128 without the need for either cipher text or plaintext. The approach also works on AES-128 implementations that use compression tables, such as Open SSL.

3.2 S-BOX and IS-BOX:

In cryptography, an S-box (substitution-box) is a basic component of symmetric key algorithms which performs substitution. In block ciphers, they are typically used to obscure the relationship between the key and the ciphertext — Shannon's property of confusion.

In general, an S-box takes some number of input bits, m, and transforms them into some number of output bits, n, where n is not necessarily equal to m. An m×n S-box can be implemented as a lookup tablewith 2^m words of n bits each. Fixed tables are normally used, as in the Data Encryption Standard (DES). But in some ciphers the tables are generated dynamically from the key (e.g. the Blowfish and the two fish encryption algorithms). One good example of a fixed table is this 6×4-bit S-box from DES (S_5):

S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

Table 3.1. S-Box Design

Given a 6-bit input, the 4-bit output is found by selecting the row using the outer two bits (the first and last bits), and the column using the inner four bits. For example, an input "011011" has outer bits "01" and inner bits "1101"; the corresponding output would be "1001".
The 8 S-boxes of DES were the subject of intense study for many years out of a concern that a backdoor — a vulnerability known only to its designers might have been planted in the cipher.

The S-box design criteria were eventually published (in Coppersmith 1994) after the public rediscovery of differential cryptanalysis, showing that they had been carefully tuned to increase resistance against this specific attack. Biham and Shamir found that even small modifications to an S-box could significantly weaken DES. There has been a great deal of research into the design of good S-boxes, and much more is understood about their use in block ciphers than when DES was released.

Any S-box where each output bit is produced by a bent function of the input bits, and where any linear combination of the output bits is also a bent function of the input bits, is a perfect S-box. The inverse S-Box is simply the S-Box run in reverse.

DISADVANTAGE:

- There are several attacks with which key from the S-Box can be retrieved easily.
- The S-Box is stored in the database and the key can be retrieved once this database is hacked.
- In IS-Box the values of S-Box are reversed and due to this if one value is known then its equivalent key value can be found.

3.3 PP-BOX:

In Partially Populated-Box the size of the S-Box is reduced from 256 bits to 128 bits. By this the possibility of finding the keys by the unauthorised person is reduced

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

and with this confusion to the S-Box is also increased. This reduction of the S-Box will increase the security of the S-Box.

DISADVANTAGE:
- ✓ Though the box size is reduced the time complexity of finding the key is increased to O(n).

3.4 IPP-BOX:
Inverse partially populated box is designed by using the values that is left in PP-Box. The values are entered into the IPP-Box are used to find the key values. The values that are left in the PP-Box can be found by using the complement of the keys that are found in the PP-Box. Due to this the time complexity is reduced from O(n) to O(2).

DISADVANTAGE:
- ✓ Though the time complexity is reduced size of the Substitution-Box get increased.
- ✓ Both the boxes use only 128-bits and the remaining memory is wasted.

IV. PROPOSED SYSTEM

In our project we propose the concept of using single S-Box for AES and the values are encrypted by using 16^{th} hexadecimal complement with a key k.

$$\text{S-Box}(i,j)=\text{ES-Box}\{(\overline{i-k}),(\overline{j-k})\}$$

Encrypted entry constitutes ES-Box (Encrypted Substitution-Box). Here in ES-Box we avoid the self invertible property.

ADVANTAGES:
- ✓ Overcomes the waste of memory by using single S-Box.
- ✓ Execution time will always be in O(1).
- ✓ Self-Invertible property is the key factor for the hackers to attack the S-Box which is prevented in our ES-Box. That makes S-Box more reliable.
- ✓ Size of the S-Box is reduced from 512(in case of IPP-Box) to 256.
- ✓ Security has been improved with the key factor k that depends on the user.

4.1 OPTIMIZED SYSTEM:

4.4.1 DIRECT S-Box:
The name itself reveals that entries are in direct form in which it becomes the door step for the hackers to attack the cryptosystem. Another possible way for the hackers who track the security key can easily hit on the database to engage hacking.

4.4.2 IS-Box:
Inverse S-Box constitutes the Self-Invertible property of S-Box
S-Box(i,j)=IS-Box
Provides security to the S-Box but increases the space and time complexities.
Time Complexity : O(2)
Space Complexity : 512 bytes

4.4.3 DYNAMIC S-Box:
Dynamic S-Box provides more reliable cryptosystem in which each instance the S-Box is created according to the blocks and input message. When keen digging about the methodology is applied, there comes the possibility of complexities towards the dynamic procedure. Applying dynamic procedure trends towards the randomness that has to be developed in receiving side leads to the hack of the cryptosystem.
Time Complexity: Time Complexity[dynamic] + Time Complexity[search]
Space Complexity: 256b

4.4.4 PP-Box:
Partially Populated S-Box is obtained by the Self-Invertible property of the S-Box
- S-Box(i,j)=S-Box(i,j)

Thus the repeated entries of that property is avoided and partially populated box alone is obtained
- If PP-Box(i,j) : Value Then PP-Box(j,i)=null
- If PP-Box(i,j) : null Then PP-Box(j,i)=Value

Case i) when the data to be fetched is present in the PP-Box
- Time Complexity : O(1)
- Space Complexity : used space(128 bytes)+wastage(128 bytes)
- Case ii) when the data to be fetched is not present in the PP-Box
- Time Complexity : O(1+n)
- Space Complexity : used space(128 bytes)+wastage(128 bytes)

4.4.5 IPP-Box:
Inverse PP-Box is implemented to overcome the time complexity of PP-Box search. The entries that are not presented in PP-Box are complements by the procedure of 16^{th} complement and stored in another table which is represented to be IPP-Box.

Thus the repeated entries of that property are avoided and all the entries are obtained through the development of IPP-Box.
- If PP-Box(i,j) : Value Then IPP-Box($\bar{i},\bar{j}$)=null
- If PP-Box(i,j) : null Then IPP-Box($\bar{i},\bar{j}$)=Value

Case i) when the data to be fetched is present in the PP-Box
Time Complexity : O(1)
- Space Complexity : used space(256 bytes)+wastage(256 bytes)
- Case ii) when the data to be fetched is not present in the PP-Box
- Time Complexity : O(2)

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

- Space Complexity : used space(256 bytes)+wastage(256 bytes)

4.4.6 ES-Box (Optimized System):

Overcomes the wastage of memory by using single S-Box (256 bytes). Execution time will always be in O(1) by which all the entries are provided in single encrypted S-Box. Self-Invertible property is the key factor for the hackers to attack the S-Box which is prevented in our ES-Box. That makes S-Box more reliable. Size of the S-Box is reduced from 512(in case of IPP-Box) to 256. Security has been improved with the key factor k that depends on the user. In our project we propose the concept of using single S-Box for AES. In this S-Box the values are encrypted by using 16th hexadecimal complement with a key k.

- S-Box(i,j)=ES-Box{($\overline{i-k}$),($\overline{j-k}$)}

Thus this encrypted entry constitutes ES-Box (Encrypted Substitution-Box).
Here in ES-Box we avoid the self invertible property.
Time Complexity: O(1)
Space Complexity: 256 bytes

V. DESIGNING OF S-BOX

In the designing of S-Box we will design the S-Box in our own methods and its designed by the sender and user and it will vary from the person to person. Here in our project the S-Box is designed by using the encrypted entries. The encrypted values are obtained by converting the given elements to hexadecimal values and doing 16th complement for the converted value. The design is done in such a manner that there is no conflicting of values.

S-Box Design:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0F	B3	55	75	D3	09	C7	1D	B1	4C	60	ED	43	7D	89	1C
1	83	4D	67	F5	18	92	44	62	C5	54	B7	2E	DA	A2	6A	BF
2	7B	6E	94	2D	B9	4E	E4	2A	57	C7	1B	AD	87	FC	02	CD
3	7E	8C	29	B1	3F	65	B6	39	7A	FA	34	E8	3C	71	A9	07
4	4F	E1	3B	AB	5A	98	28	DC	8B	E6	CA	49	85	FF	1A	F8
5	07	5C	6E	D5	19	BD	46	C2	3A	6A	59	81	27	8E	E3	42
6	9C	6Y	76	26	AD	37	72	11	BB	08	10	78	DE	47	EB	5D
7	3E	35	A3	5E	CF	68	0B	F2	25	D4	41	B4	20	F0	38	A5
8	51	52	D0	F3	33	E0	C8	24	95	0C	9E	FD	1F	17	94	C3
9	0D	2F	AE	50	DF	4B	A7	58	32	EE	21	CB	9F	23	E9	16
A	F6	22	80	91	AF	15	B6	A6	69	D8	48	FE	E7	31	7F	80
B	6D	C0	DB	EF	F7	6B	30	B5	56	93	AA	03	EA	F4	A1	86
C	04	CC	90	15	C4	BE	A8	C9	82	9B	05	8F	C6	7C	68	53
D	4A	6F	E5	DD	12	0E	D9	CE	BA	A4	9D	88	79	63	5F	66
E	84	74	3D	F9	F1	D6	B2	AC	EC	73	61	5B	06	01	2C	99
F	96	00	2B	70	77	8A	E2	FB	97	B0	14	36	40	45	D2	13

Fig.5.1 Design of S-Box

5.1 BYTE SUSTITUTION

The given message is converted into hexadecimal value and the hexadecimal values are entered into a 4*4 matrix. Taking the first value of the hexadecimal value as row and second as column the values are entered into the S-Box and are retrieved from the S-Box.

5.2 ROW SHIFT:

The entered hexadecimal values are considered in 4*4 matrix and are shifted in following manner.,

\0th Row => No Shift 1st Row => 1st Shift
3rd Row=> 2nd Shift 4th Row=> 3rd Shift

5.3 COLUMN MIX:

In column mix generally random values are used as the key here in this project we use block size as random key values and these key values are used for column mix.

5.4 ROUND KEY ADDITION:

In round key addition generally we use original key for encryption. Here we use the original key, based on the original message or input message. We use 10 rounds for encryption in 128-bits.

After this step at final we obtain an encrypted output value. This encrypted message is transferred by the sender to receiver

5.5 REVERSE ROW SHIFT:

In encryption technique we use left row shift but in decryption technique we use reverse row shift technique (i.e) Right Shift is done. The remaining decryption works are same as the encryption technique.

5.6 DATABASE CONNECTIVITY FOR S-BOX:

In database connectivity we use MYSQL with MYSQL server for the database connectivity. We connect S-Box with the coding by using database connectivity. Queries are written for database access.

VI. CONCLUSION

Advanced Encryption Standard is one of the secured and reliable systems but there exists known and side-channel attacks. These attacks are based on the brute force algorithm for key generation and possible collective key analysis. We build security for AES in the thought that key has already hacked.

The S-Box is encrypted in database in which our proposed system is known to be Encrypted S-Box (ES-Box) .Thus we conclude that even when the key is hacked, our system provides reliable to the information with effective utilization of time and space complexities.

In Future Randomize round key and modify this in each round and also to develop the decryption technique. Known and side-channel attacks which should destructed by means of symmetric key methodology.

DOI 10.17148/IJARCCE.2016.51031

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 10, October 2016

REFERENCES

[1]. Eltayeb Abuelyaman and Abdul-Aziz Alsehibani "A Real Time S-Box construction using arithmetic modulo prime number" The International Journal of Computer Science and Network Security, Vol. 5, No. 6, pp 354-360, December 2007.
[2]. Rijmen, V. "Security and Implementation of the AES" 2nd International Workshop on the state of the art in cryptology and new challenges ahead, Warsaw, Poland, Thursday, May 13th, 2004.
[3]. James Foti* "Status of the Advanced Encryption Standard (AES) Development Effort" National Institute of Standards and Technology (NIST), MD 20899-8930, March 2003.
[4]. Daemen,J. and Rijmen, V. , "AES Proposal: Rijndael," Document vers on 2, Date: 03/09/99. Retrieved on October 20, 2005.
[5]. Harvey, I."The Effects of Multiple Algorithms in the Advanced Encryption Standard", nCipher Corporation Ltd., 4'Th January 2000.
[6]. Guy, R. K. "Euler's Totient Function," "Does Properly Divide ," "Solutions of ," "Carmichael's Conjecture," "Gaps Between Totatives," "Iterations of and ," "Behavior of and ." §B36-B42 in Unsolved Problems in Number Theory, 2nd ed. New York: Springer-Verlag, pp. 90-99, 1994.
[7]. Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., Ferguson, N. "Twofish: A 128-Bit Block Cipher", 15'Th June, 1998
[8]. Anderson, R., Biham, E. and Knudsen, L. ," The Case for Serpent" , 24th March 2000
[9]. Ronald L. Rivest1, M.J.B. Robshaw2, R. Sidney2, and Y.L. Yin2, "The RC6 Block Cipher", August 20, 1998
[10]. IBM MARS Team, "MARS and the AES Selection Criteria", May 15, 2000
[11]. Daemen, J. and Rijmen, V. "AES Proposal: Rijndael " Document version 2 1999 – May
[12]. Swankoski,, E.J., Brooks, R.R., Narayanan, V., Kandemir, M., Irwin, M.J "A Parallel architecture for Secure FPGA Symmetric Encryption" , 2004 .
[13]. Harvey, I.,"The Effects of Multiple Algorithms in the Advanced Encryption Standard", nCipher Corporation Ltd., 4'Th January 2000 Retrieved on November 6, 2005