ISSN: 2321-7782 (Online)
Volume 2, Issue 4, April 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

# *Implementation of sub key generation algorithms*

**Jyotika Chandra**[1]
II year M. Tech Student, Dept. of CSE
FET, Mody University of Science & Technology
Lakshmangarh, Rajasthan - India

**Dr. Prema K.V**[2]
Professor & Head, Dept. of CSE
FET, Mody University of Science & Technology
Lakshmangarh, Rajasthan - India

***Abstract:*** ***In this project we describe techniques to generate the sub keys in DES, IDEA, and AES algorithms. These algorithms have higher resistance against brute-force attack, differential cryptanalysis and linear cryptanalysis. Furthermore, comparison of the different sub key generation algorithms is done on the basis of sub key generation time, security. DES encrypts 64 bits block using key size of 56 bits. It uses 16 rounds which perform identical operations in series of steps and produces 64 bit output. With the key of 128 bits, IDEA is far more secure than the widely known DES based on 56 bit key. The block cipher IDEA operates with 64-bit plaintext and cipher text blocks and is controlled by a 128-bit key. IDEA algorithm has flexible characteristics to be compressed or expanded by varying the block length, the key length, total number of rounds, and the bit positions to be shifted for sub-key generation. AES is a symmetric block cipher. The block and key in AES can be chosen independently from 128, 160, 192, 224, 256 bits and need not be the same. These algorithms provide better performance concerning Avalanche effect criteria. I have implemented sub key generation of these algorithms in java and then performance analysis of algorithms is done.***

***Keywords: Cryptography, Network Security, Sub Key Generation, Performance analysis.***

## I. INTRODUCTION

Why we use different sub key algorithm and not same algorithm? The reason is different algorithms have different security involved, different time complexity. In sub key generation process, the sub keys change slightly with every pair of sub key generated. This is primarily to protect against any attack of sub key generation. All these encryption algorithms are used to provide security to data with the help of different sub keys generated.

### 1.1 IDEA

Idea works on 64-bit plain text blocks. The key is longer and consist of 128 bits. The 64 bit input plain text block is divided into four portions of plain text (each of size 16 bits), say P1 to P4. Thus P1 to P4 are inputs to the first round of algorithm and there are 8 such rounds. The output of first round is input to second round. Similarly output of round second round is input to third round, and so on. In each round six sub keys are generated from the original key. Each of the sub keys consists of 16 bits. For first round we will have keys K1 to K6. For second round we will have keys K7 to K12. Finally for eighth round we will have keys from K43 to K48. The final step consists of output transformation, which uses just four sub keys (K49 to K52). The final output produced is the output produced by output transformation, which is four blocks of cipher text C1 to C4. These are combined to form final 64 bit cipher text block. In this algorithm addition is done modulo $2^{16}$ and multiplication is done modulo $2^{16}+1$. Idea employs the technique of key shifting. Key generation process is as follows-

Using 25-bit circular left shift operation on the original key, we produce other subsequent sub-keys, used in different rounds. For instance, among the total no. of 52 keys- Sub-key K1 is having first 16bits of the original key, sub-key K2 is having the next 16 bits, and so on till sub-key K6; i.e., for ROUND1, sub-keys K1 to K6 use first (16x6=) 96 bits of the original cipher key. In ROUND2, sub-key K7 & K8 take the rest of the bits (bits 97 to 128) of the original cipher key. Then we perform circular left shift (by 25bits) operation on the original key. As a result the 26th bit of the original key shifted to the first position and

becomes the first bit (of the new shifted key) and the 25th bit of the original key, moves to the last position and becomes the 128th bit (after first shift). This process continues till ROUND8.[3]

**Sub keys and corresponding bit position:[3]**

| Bit positions | Sub- keys |
|---|---|
| 1-96 | ROUND1( K1 to K6); ROUND5(K25 to K30);& OT (K49 to K52 Bit position 1-64 |
| 97-128 & 1-64 (of shifted key) | ROUND 2(K7 to K12) & ROUND(K31to K36) |
| 65-128 & 1-32 (of shifted key) | ROUND 3(K13to K18) & ROUND7(K37to K42) |
| 33-128 | ROUND 4(K19 to K24) & ROUND(K43 to K48) |

**1.2 AES**

AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. However, AES is quite different from DES in a number of ways. The algorithm Rijndael allows for a variety of block and key sizes and not just the 64 and 56 bits of DES' block and key size. The block and key can in fact be chosen independently from 128, 160, 192, 224, 256 bits and need not be the same. However, the AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys - 128, 192, 256 bits. Depending on which version is used, the name of the standard is modified to AES-128, AES-192 or AES-256 respectively. As well as these differences AES differs from DES in that it is not a feistel structure.

**Inner Workings of a Round**

The algorithm begins with an **Add round key** stage followed by 9 rounds of four stages and a tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of it's counterpart in the encryption algorithm. The four stages are as follows:

1. Substitute bytes
2. Shift rows
3. Mix Columns
4. Add Round Key

The tenth round simply leaves out the **Mix Columns** stage. The first nine rounds of the decryption algorithm consist of the following:

1. Inverse Shift rows
2. Inverse Substitute bytes
3. Inverse Add Round Key
4. Inverse Mix Columns

**AES Key Expansion**

The AES key expansion algorithm takes as input a 4-word key and produces a linear array of 44 words. Each round uses 4 of these words as shown in figure 7.2. Each word contains 32 bits which means each sub key is 128 bits long. Pseudocode for generating the expanded key from the actual key.

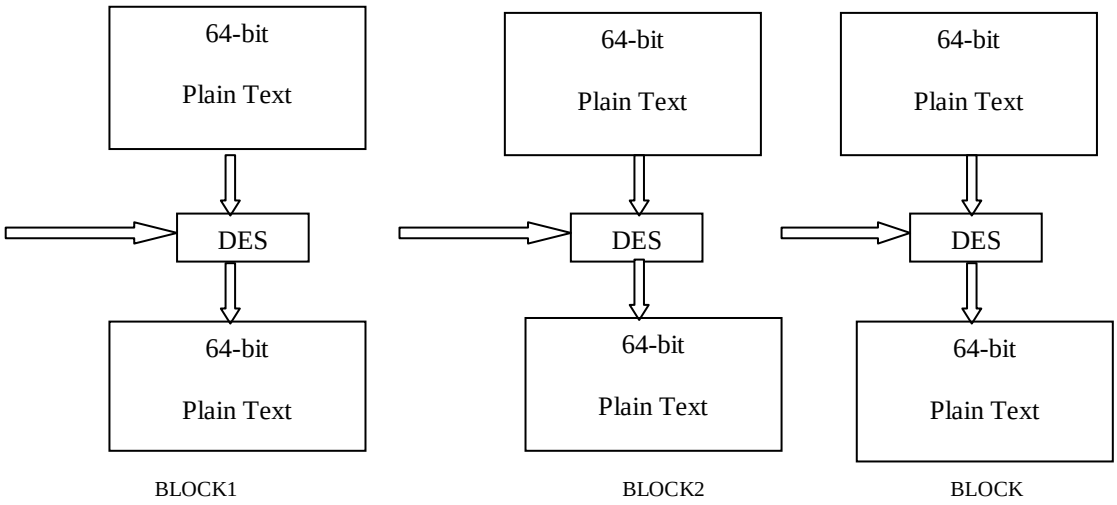
Key expansion (byte key[16], word w[44])

```
{
  Word temp

   for (i=0; i<4;i++)

 {
W[i] = (key[4*i+1],key(4*i+2),key[4*i+3]);
}
for(i=4;i<44;i++)

{
temp=w[i-1]; if(I mod 4 =

0)

temp=Subword(RotWord(t

emp))^ Rcon[i/4];

w[i]=w[i-4]^temp;}}
```

**1.3 DES**

Data Encryption Standard (DES) also known as Data Encryption Algorithm. DES is a block cipher. It encrypts data in block size of 64 bit each. That is, 64 bits of plain text goes as input to DES, which produces 64 bits of cipher text. DES uses a 56-bit key. That is, bit position 8, 16,24,32,40,48,56,64 are discarded.




**Conceptual working of DES**

Steps in DES:

1. In first step, the 64-bit plain text is handed over to an Initial Permutation(IP) function .

2. The initial permutation is performed on plain text.

3. Next, the initial permutation (IP) produces two halves of the permuted block; say Left Plain Text(LPT) and Right plain Text(RPT)

4. Now each of LPT and RPT go through 16 rounds of encryption process.

5. In the end, LPT and RPT rejoined and a Final Permutation (FP) is formed on the complete block.

6. The result of this process produces 64 bit cipher.

Each of the 16 rounds consists of following steps:

**1.3.1** .Key Transformation

**1.3.2** Expansion permutation

**1.3.3** S-box substitution

**1.3.4**. P-Box permutation

**1.3.5**. XOR and Swap

**1.3.2 KEY TRANSFORMATION:** Initial 64- bit key is transformed into 56 bit key by discarding every $8^{th}$ bit of initial key. Thus for each round, a 56 bit key is available. From this 56-bit key, a different 48 bit sub key is generated during each round using a process called a key transformation. For this, the 56 bit key is divided into two halves, each of 28 bits. These halves are circularly shifted left by one or two positions, depending on the round. For example, if round number is 1, 2, 9 or 16, the shift is done by one position. For other rounds, the circular left shift is done by two positions.

**1.3.2 Expansion permutation:** During expansion permutation, the RPT is expanded from 32 bits to 48 bits. Besides increasing the bit size from 32 to 48 bits, the bits are permuted as well.

**1.3.3 S box substitution:** S box substitution is the process that accepts 48 bit input from the XOR operation involving the compressed key and expanded RPT and produces a 32 bit output using the substitution technique. The substitution is performed by eight substitution boxes (also called as S boxes). Each of the substitution boxes has 6 bit input and 4 bit output. The 48 bit input block is divided into 8 sub blocks(each containing 6 bits) and each sub block is given to S- box. The S box transforms the 6 bit input into a 4 bit output.

**1.3.4 P box permutation:** The output of S-box consists of 32 bits. These 32 bits are permuted using P box. This straight forward permutation mechanism involve simple permutation(i.e replacement of each bit with another bit), specified in P box table, without any expansion or compression).

**1.3.5 XOR and Swap:** The Left half portion of initial 64 bit plain text block (i.e LPT) is XORed with the output produced by P box permutation. The result of this XOR operation becomes new right half.(i.e RPT). The old right half (i.e RPT) becomes new left half, in a process of swapping.[7]

Triple DES is DES three times. It comes in two flavors: One that uses three keys and the other that uses two keys. Triple DES is not susceptible to meet in the middle attack, unlike double DES.[7]

## II. CONCLUSION

We have implemented sub key generation of IDEA, DES and AES algorithm in Java. Then we have compared this algorithm on the basis of security, time complexity i.e performance analysis is done. We have designed one front end for selection of algorithm ie which sub key algorithm user wants to choose.

To break IDEA and AES 2128(1038) encryption operation would be required. So it is very difficult to break IDEA and AES. Out of these three algorithm AES algorithm is better. After that we have developed General Sub key Generation Algorithm that is as secure as these algorithms.

## References

1. Mandal B.K. , Bhattacharyya.D, Bandopadhyay S.K., "Designing and performance analysis of a proposed symmetric cryptrography," "in Computer Systems and Network Technologies(CSNT)" 2013 International Conference, 2013 © IEEE. Doi:10.1109/CSNT.2013.101
2. Ramesh.A, Suruliandi.A," Performance analysis of encryption algorithms for information security," "in Circuits Power and computing technologies(ICCPCT)"2013 International Conference, 2013 © IEEE. Doi:10.1109/ICCPCT.2013.6528957
3. Sandipan Basu,"International Data Encryption Algorithm(IDEA)-A typical illustration," Journal of global research in Computer Science, Vol 2, no. 7,7 July 2011
4. William Stallings, Cryptography and Network Security. 4th ed. Pearson education Inc, 2006
5. www.schneier.com/blowfish.htm
6. www.media-crypt.com/
7. Atul Kahate, Cryptography and Network Security, 2nd ed. Tata McGraw Hill education Pvt Ltd.,2003