Intensifying Shared-Key Based Cryptography Using Rbits

Penchalaiah P[1], Dr. Ramesh Reddy K[2]

[1]Research Scholar, Department of Computer Science, Vikrama Simhapuri University, Nellore
Andhra Pradesh, India

[2]Asst.Professor, Department of Computer Science, Vikrama Simhapuri University, Nellore
Andhra Pradesh, India

[1]Penchal.caliber@gmail.com
[2]Drkrreddy05@gmail.com

Abstract— a weak key makes the cipher to behave in some undesirable way. Even though weak keys usually represent a very small fraction of the overall keyspace it is desirable for a cipher to have no weak keys because weak keys give clues to break the cipher. Weak keys are the keys those make the same sub-key to be generated in more than one round. Generation of same sub-keys for more than one round leads to reduce of cipher complexity. So it is must to avoid weak keys at key generation component to continue with the same temper of cipher complexity. In this paper, we are extending Rbits to DES family and IDEA algorithms. We are introducing Rbits as a key supplier or generator for the existing algorithms to avoid weak key sub-key generation. In brief Rbits is a kind of symmetric key encryption scheme. It aims to provide confidentiality, authentication for data with the use of random bits. These random bits itself forms sub-keys on demand and problem context.

Index Terms— Weak key, Encryption, Decryption, Random Numbers, Rbits, DES, IDEA

I. Introduction

With the advancements in networks and storages technology, organizations are expectant to collect great volumes of sensitive data. The presence of the immense amount of sensitive information that is being stored in media or transmitted over the networks has imposed various threats to user confidentiality and privacy.

Feistel proposed the idea of product cipher which is composition of several different functions in sequence such a way that the produced ciphertext is cryptographically stronger. Actually, this is a proposal of Claude Shannon [1] to develop a product cipher. In this approach, numerous substitutions and permutations are used to develop strong block cipher. The working model of Feistel called Feistel structure, in which encryption function 'E' takes plaintext blocks as input. The input is divided into two halves, L (Left) and R (Right). Then, the two halves (L and R) undergo 'n' rounds of processing and finally combine to produce ciphertext blocks as output. Each round R_i has as inputs as its previous round outputs (L_{i-1} and R_{i-1}), as well as a unique sub-key K_i , derived from the key K and all rounds have the same processing. Transformation is performed on the 'L', by applying a round function 'E' to 'R', which is further bitwise XORed with the 'L'. Each round function has common structure but each round is parameterized by unique sub-key K_i.

II. DES And Weak Keys

DES stands for Data Encryption Standard [2], developed by IBM in 1974 which Feistel structure. It is a 16-round cipher with block size of 64 bits. A 64 bit plaintext blocks are handed over to the initial permutation (IP) function which produces two halves; say L and R and followed by 16 rounds, each with its own unique key. The inner working of DES single round [3] along with key supply is depicted in the figure 1.

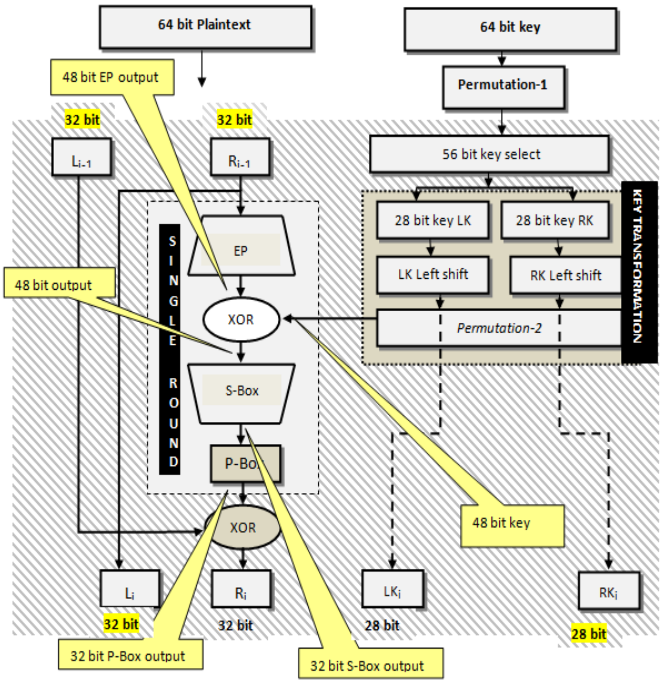

Figure 1: DES inner working

From the figure 1 each round contains the following functionalities in sequence:

a. Key transformation/Generation (KT)
b. Expansion Permutation (EP)
c. S-Box Substitution (SB)
d. P-Box Permutation (PB)
e. XOR

A. Key Schedule

A 56 bit key is divided into two 28-bit halves. In successive rounds, both halves are rotated left by one or two bits. 48 bit key as sub-key are selected by permutation choice i.e. 24-bits from the left half, and 24-bits from the right. The generated 48-bit key is used as a key for encryption in a single round.

B. Weak Keys

In DES, there might be inherited linear factors which cannot be derived directly, but under our assumptions about DES weak key for the whole cipher would consist of sequences of linearity for the individual rounds of DES. David Cham work [4] defines principle to find linear factors on block ciphers. Block cipher is said to have a linear factor if, *∀{p,k}, ∃{k1,k2,..kn}* where *p, k* are plaintext , key respectively and k_1,k_2…..k_n are fixed non-empty set of key bits whose simultaneous complementation results the XOR of a fixed non-empty set of ciphertext bits unchanged. DES consists of 16 rounds of the form:

$$Li+1=Ri,$$

$$Ri+1=Li\oplus F(Ri,Ki)$$

All rounds are identical except for the round subkey K_i. The subkeys K_i are derived from the encryption key ‘K’ using the DES key schedule, which takes the form.

$$\mathrm{K_L} \parallel \mathrm{K_R} = \mathrm{P1(K)}$$

$$\mathrm{K_i} = \mathrm{P2}(\mathrm{K_L} \lll \mathrm{n_i}, \mathrm{K_R} \lll \mathrm{n_i})$$

where K_L and K_R are the left and right halves of the permuted key P1(K), the functions P1 and P2 are fixed maps that basically just shuffle the bits around, and ⋘ni denotes bit rotation by the fixed number of positions ni.

The significant thing is that, if K_L and K_R each consist of all one or all zero bits, then rotating them leads no effect, and so all the subkeys K_i generated will be same as K_j which indirectly implies that all rounds of DES encryption uses a single key. Since there are two choices for each of K_L and K_R this gives us a total of 2^2 four weak keys. The four weak keys (64 bits) value (with parity bits) in hex decimal form is:

- 0101 -0101 -0101 -0101
- 1F1F -1F1F -0E0E -0E0E
- E0E0 -E0E0 -F1F1 -F1F1
- FEFE -FEFE -FEFE -FEFE

Actual keys (56 bits) with-out parity bit which is derived from eliminating every 8th bit from binary representation is:

- 0000000 – 0000000
- 0000000 – FFFFFFF
- FFFFFFF – 0000000
- FFFFFFF – FFFFFFF

The four 4 keys set { K_L || K_R } : {1^{28} ||1^{28}}, {0^{28} || 0^{28}}, {1^{28} || 0 ^{28}} {0^{28}|| 1^{28}} are called weak keys.

DES also exhibits semi-weak keys, for which encryption with one of the keys in the pair is equivalent to decryption with the other [5][6]. Basically, these are the keys for which the bit patterns of K_L and K_R repeat with a period of two, i.e. alternative 1 and zero vice versa. Here is the DES Semi weak Key Pairs.

- 01FE- 01FE- 01FE- 01FE and FE01- FE01- FE01- FE01
- 1FE0- 1FE0- 0EF1- 0EF1 and E01F- E01F- F10E- F10E
- 01E0- 01E0- 01F1- 01F1 and E001- E001- F101- F101
- 1FFE- 1FFE- 0EFE- 0EFE and FE1F- FE1F- FE0E- FE0E
- 011F- 011F- 010E- 010E and 1F01- 1F01- 0E01- 0E01
- E0FE- E0FE- F1FE- F1FE and FEE0- FEE0- FEF1- FEF1

III. IDEA AND WEAK KEYS

The initials version of IDEA (International Data Encryption Algorithm) was launched in 1990 and called as PES (Proposed Encryption Standard). The working procedure of IDEA is [7]: IDEA takes the input plain text of 64 bits and divides into 4 blocks say B1 to B4 each of size 16 bits. The blocks undergo eight rounds and followed by an output transformation in sub-keys. In each round the following operations are performed.

(a) In each round 6 sub-keys (K_i where I = 1 to 6 for a single round) are generated from the original key **K**. Each of the sub-keys consists of 16-bits since of the block size is 16 bits. These six sub-keys are applied to four input blocks B1 to B4.Thus for first round, we have 6 keys say k1 to k6; for second round , we have k7 to k12.Finally for eight round we have keys k43 to k48.

(b) Plaintext blocks are getting Multiplied, added and XORed with sub keys.

A. Key Schedule

In brief the key schedule of IDEA divides the 128 bit key into 8 round keys, each 16 bit long. This key schedule is fully linear and exhibits patterns in the key even in the last round keys with almost no change [8].

B. Weak Keys in IDEA

A class of weak keys yielding liner factor and characteristics with probability 1 has been found for the block cipher algorithm IDEA. Here keys are weak in the sense that it takes only a very small amount of effort to detect their use. In IDEA, multiplication by -1 inherits the linearity properties of the addition modulo 2. The use of multiplicative sub keys with value 1 or - 1 give rise to linear factors in the round function [8][9]. For a class of 223 keys IDEA exhibits a linear factor, linear factor is a linear equation in key. For a certain class of 235 keys the cipher has a global characteristic with probability 1. For another class of 251 keys only two encryptions and solving a set of 16 nonlinear Boolean equations with 12 variables is sufficient to test if the used key belongs to this class [9][10].

IV. Rbits

Rbits [11] can be view as a tuple {SAlgo, RGAlgo} a pair of algorithms. The algorithm SAlgo produces the initial seed; RGAlgo takes the initial seed as input and returns a sequence output blocks Ob1, Ob2, . . . pseudorandom block.

$$S0 \leftarrow SAlgo$$

By iterating:

$$(Oj,\ Si) \leftarrow RGAlgo\ (Si-1);\ \ for\ i \geq 1.$$

These random bits are generated based on set of shared parameters where these parameters are selected on certain complex criteria. These random bits itself forms sub-keys on demand and problem context which are continuously supplied to encryption functions.

A. *Core Parameters Generation*

For key generation, Rbits uses cryptographically proved and secured BBS algorithm. The produced random bits are used to construct sub-keys. The following pseudo code is used to select the corekey and parameter. The SAlgo selects CK, P, $\{CK \leftarrow Z_{0,} P \leftarrow n\}$, where ‘$Z_0$’ is the common core-key and ‘n’ is parameter ‘P’ [11]. Once the two entities selected the following pseudo code is used to generate sub-keys.

B. *Sub-keys Generation (RGAlgo)*

The below pseudo code generates multiple preset size of sub-keys to encrypt plaintext of size msgLen [11][12].

```
loop i=1 to msgLen
        $CKi = (CKi - 1)^2$  mod  n
        bi= LSB of CKi  //i.e.  CKi  mod 2
                kj= kj | bi
         if (preset keySize) then
                increment  j
endloop
```

V. Rbits As A Sub-Key Generator

Even though weak keys usually represent a very small fraction of the overall keyspace it is desirable for a cipher to have no weak keys because weak keys give clues to break the cipher. Weak keys are the keys those make the same sub-key to be generated in more than one round. Generation of same sub-keys for more than one round leads to reduce of cipher complexity. So it is must to avoid weak keys at key generation component to continue with the same temper of cipher complexity [13][14][15].

Rbits key generation component takes a seed ‘S’ as input and generates required number of bit as key demand. By replacing the key generation component in figure 2 with Rbits key generation component, we can further improve the performance and complexity of DES.

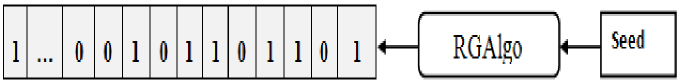

Figure 2: Rbits key Generation Component

Rbits key generation component can be applied to DES family ciphers, by supplying the strong sub-keys the complexity and security of ciphers can be improved.

A. *DES and Rbits*

Unlike DES key schedule in which 16 sub-keys are generated from 56-bit key by permutations, for each round a new 48-bit sub-key is generated and used for encryption. Generation of 16 independent sub-keys for 16 rounds of DES eliminates weak key problem of DES and is shown in figure 3.

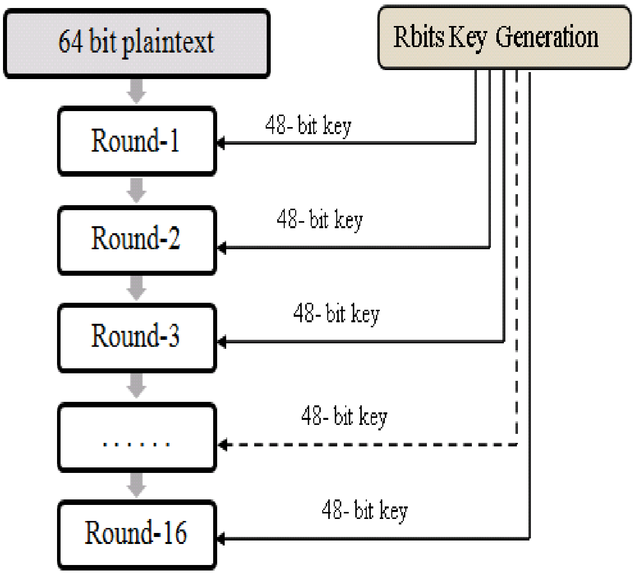

Figure 3: Rbits key supply for DES

B. *IDEA and Rbits*

Similar to DES, it is possible to eliminate the weak key problems of IDEA by replacing the key schedule by Rbits. An Rbits key generation technique can be applied to IDEA as shown in figure 4 and can avoid liner and pattern in keys used for encryption.

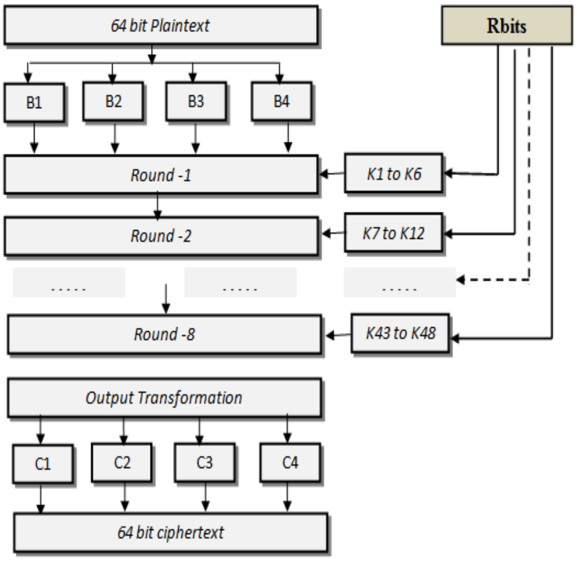

Figure 4. Rbits key supply for IDEA

VI. CONCLUSION

It is possible to eliminate the weak key problems of ciphers by replacing the key schedule by Rbits. An Rbits key generation technique can be applied to existing ciphers as shown above and can avoid liner and pattern in keys used for encryption. Rbits generates strong, random and unpredictable sub-keys [12]. Rbits key generation component is not just limited to only DES or IDEA algorithms. The cipher which requires or takes multiple sub-keys for encryption operations can use the services of Rbits key generation.

REFERENCES

[1] C. E. Shannon, "Communication theory of secrecy Systems", Bell System Technical Journal, vol. 28, no. 4, pp. 656–715, 1949.

[2] Rajdeep Bhanot and Rahul Hans, “A Review and Comparative Analysis of Various Encryption Algorithms”, *International Journal of Security and Its Applications (SCOPUS)*, Vol. 9, No. 4, pp. 289-306, 2015.

[3] Nikita Arora, and Yogita Gigras, “Block and Stream Cipher Based Cryptographic Algorithms:A Survey”, *International Journal of Information and Computation Technology*, Vol. 4, Number 2, pp. 189-196 , 2014.

[4] D. Chaum and J.-H. Evertse, "Cryptanalysis of DES with a reduced number of rounds," *Advances in Cryptology* — CRYPTO ’85 Proceedings, *Springer*, pp. 192–211.

[5] C. Sanchez-Avila and R. Sanchez-Reillol, "The Rijndael block cipher (AES proposal) : a comparison with DES," Proceedings *IEEE* 35th Annual 2001 International Carnahan Conference on Security Technology, pp. .229-234, 2001.

[6] M. M. Alani, "DES96 - improved DES security," 2010 7th International Multi- Conference on Systems, *Signals and Devices*, pp. 1-4, 2010.

[7] Osama Almasri and Hajar Mat Jani, “Introducing an Encryption Algorithm based on IDEA”, *International Journal of Science and Research.* Vol. 2/9, pp. 335-339, 2013.

[8] J. Daemen, R. Govaerts, and J. Vandewalle, "Weak keys for IDEA," *Advances in Cryptology* — CRYPTO’ 93, pp. 224–231, 1994.

[9] Alex Biryukov, Jorge NakaharaJr, Bart Preneel, Joos Vandewalle, “New Weak-Key Classes of IDEA”, *Springer*, Information and Communications Security, pp. 315-326, 2002.

[10] Philip Hawkes, “Differential-linear weak key classes of IDEA”, Springer, DOI: 10.1007/BFb0054121, 2006.

[11] Penchalaiah P, Ramesh Reddy K, “Random Multiple Key Streams for Encryption with Added CBC Mode of Operation*”, Perspectives in Science journal, (ELSEVIER)*, Volume 8, pp.57-62, April 2016.

[12] Penchalaiah P, Ramesh Reddy K, “Secure and Cost Effective Cryptosystem Design Based on Random Multiple Key Streams”, *Journal of Information Security Research, DIFR Publisher, (SCOPUS indexed publisher)*, Volume 7, Number 1, pp. 29-40, March 2016.

[13] Mohamed Ahmed Abdelraheem, Andrey Bogdanov, and Elmar Tischhauser, “Weak-Key Analysis of POET”, *International Association for Cryptologic Research*, pp. 1-10, 2014.

[14] Bouman N.J., Fehr S. “Secure Authentication from a Weak Key, without Leaking Information”. *Advances in Cryptology* – EUROCRYPT 2011.vol.6632. Springer, 2011.

[15] Yin, R., Wang, J., Yuan, “Weak key analysis for chaotic cipher based on randomness properties*”,J. et al. Sci. China Inf. Sci*. doi:10.1007/s11432-011-4401-x, 2012.

[16] Weak keys’ for DES," 2016. [Online]. Available: http://crypto.stackexchange.com/questions/12214/can-you-explain-weak-keys-for-des.

[17] SSLeay0.9.0bdocs,”[Online].Available: http://www.umich.edu/~x509/ssleay/des-weak.html