Volume 2, Issue 5, May 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Study and Performance Analysis of IDEA with Variable Rounds

Kamal Deep Sharma*, Harsh K Verma, Ashish Kumar
Department of Computer Science and Engineering
Dr B.R. Ambedkar National Institute of Technology
Jalandhar- 144011, Punjab (India)
Kunalattri4u@gmail.com

Abstract— ***IDEA is a 64-bit block cipher with 128-bit keys introduced by Lai and Massey in 1991. IDEA is one of the most widely used block ciphers, due to its inclusion in several cryptographic packages, such as PGP. Since its introduction in 1991, IDEA has withstood extensive crypt- analytic effort, but no attack was found on the full (8.5-round) variant of the cipher. In 2006 Eli Biham, Orr Dunkelman and Nathan Keller present the first known attack on 6-round IDEA faster than exhaustive key search. The attack exploits the weak key-schedule algorithm of IDEA, and combines Square-like techniques with linear cryptanalysis to increase the number of rounds that can be attacked. In this paper we are analyzing the performance of IDEA algorithm by making its rounds variable. By this we are trying to increase the security of IDEA algorithm.***

Keywords*— *Cryptography, Block Cipher, IDEA, IPES, PES.

I. INTRODUCTION

The International Data Encryption Algorithm (IDEA) is a 64-bit, 8.5-round block cipher with 128-bit keys proposed by Lai and Massey in 1991 [1]. IDEA (International Data Encryption Algorithm) is a block cipher which ever named PES(Proposed Encryption Standard), then the name changed to IPES(Improved PES). Due to its inclusion in several cryptographic packages, such as PGP, IDEA is one of the most widely used block ciphers. Since its introduction, IDEA resisted intensive cryptanalytic efforts [2,3,4,5,6,7,8,9]. The cipher is based on the design concept of \mixing operations from different algebraic groups". The required confusion is achieved by successively using three different group operations on pairs of 16-bit sub blocks and the cipher structure was chosen to provide the necessary diffusion. The cipher is so constructed that the deciphering process is the same as the enciphering process once the decryption key sub blocks have been computed from the encryption key sub blocks. The cipher structure was chosen to facilitate both hardware and software implementations.
In this paper we are modifying the IDEA Algorithm by changing the constant 8 rounds into variable (n-1) rounds. User now has to pass two arguments:

1. Key.
2. No. of rounds.

The cipher is described in Section 2, 3 and 4 which contains Encryption, key Scheduling and decryption. Then next section shows the comparisons among the algorithms of different rounds based on memory and execution time. Finally, Section 6 summarizes the paper.

II. DESCRIPTION OF ENCRYPTION ALGORITHM

This new IDEA encrypts a 64-bit block of plaintext to 64-bit block of cipher text. It uses a 128-bit key. The algorithm consists of (n-1) identical rounds and a "half" round final transformation The computational graph of the encryption process is shown in Fig.1. The process consists of "n-1" similar rounds followed by an output transformation. The complete first round and the output transformation are depicted in Fig.1.

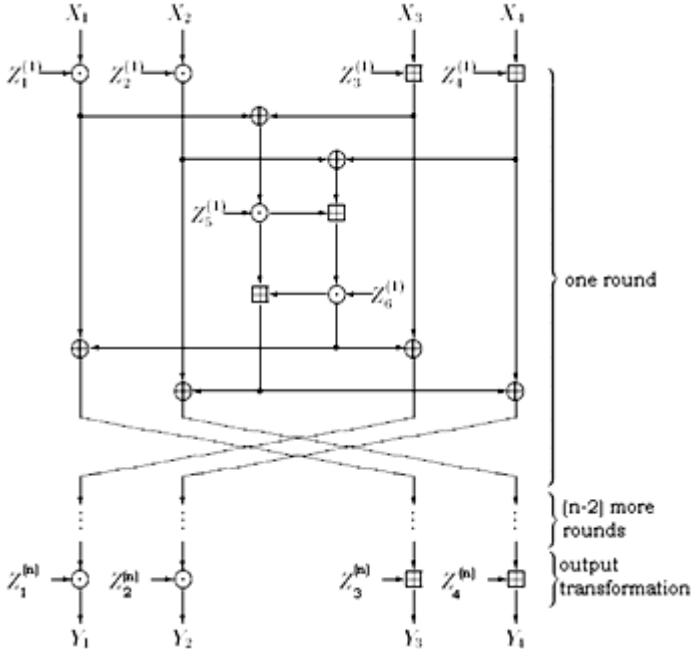

Figure 1: Computational graph for encryption.

X_i: 16-bit plaintext sub block

Y_i: 16-bit cipher text sub block

$Z_i^{(n)}$: 16-bit key sub block

$\oplus$: Bit-by-bit exclusive-OR of 16-bit sub blocks

$\boxplus$: Addition modulo 2^{16} of 16-bit integers

$\odot$: Multiplication modulo $2^{16} + 1$ of 16-bit integers with the zero sub block corresponding to 2^{16}

In the encryption process shown in Fig.1, three different group operations on pairs of 16-bit sub blocks are used, namely,

- bit-by-bit exclusive-OR of two 16-bit sub blocks, denoted as $\oplus$
- addition of integers modulo $2^{\wedge 16}$ where the sub block is treated as the usual radix-two representation of an integer, the resulting operation is denoted as $\boxplus$
- multiplication of integers modulo $2^{\wedge 16} + 1$ where the sub block is treated as the usual radix-two representation of an integer except that the all-zero sub block is treated as representing $2^{\wedge 16}$; and the resulting operation is denoted as $\odot$

There are $2^{\wedge 16}$ possible 16-bit blocks: 0000000000000000, ..., 1111111111111111, which represent the integers 0, ..., $2^{\wedge 16} - 1$. Each operation with the set of possible 16-bit blocks is an algebraic group. Bitwise XOR is bitwise addition modulo 2, and addition modulo $2^{\wedge 16}$ is the usual group operation. Some spin must be put on the elements – the 16-bit blocks – to make sense of multiplication modulo $2^{\wedge 16} + 1$, however. 0 (i.e., 0000000000000000) is not an element of the multiplicative group because it has no inverse, but by thinking of the elements of the group instead as 0000000000000001, ... 1111111111111111, 0000000000000000, which now represent the integers 1, ..., $2^{\wedge 16} - 1$, $2^{\wedge 16}$, everything works for multiplication. $2^{\wedge 16} - 1$ mod $2^{\wedge 16} + 1$, and 0000000000000000 is its own inverse under multiplication modulo $2^{\wedge 16} + 1$. For a description of IDEA, we follow Schneier [10], who breaks the encryption algorithm into fourteen steps.

For each of the (n-1) complete rounds, the 64-bit plaintext block is split into four 16-bit sub-blocks: X_1, X_2, X_3, X_4. The 64-bit input block is the concatenation of the sub blocks: $X_1 \| X_2 \| X_3 \| X_4$, where $\|$ denotes concatenation. Each complete round requires six sub keys. The 128-bit key is split into eight 16-bit blocks, which become eight sub keys. The first six sub keys are used in round one, and the remaining two sub keys are used in round two. We will discuss the generation of the remaining keys in the next section. Each round uses each of the three algebraic operations: bitwise XOR, addition modulo $2^{\wedge 16}$ and multiplication modulo $2^{\wedge 16} + 1$.

Here are the fourteen steps of a complete round (multiply means multiplication modulo $2^{\wedge 16} + 1$, and add means addition modulo $2^{\wedge 16}$):

1. Multiply X_1 and the first sub key Z_1.
2. Add X_2 and the second sub key Z_2.
3. Add X_3 and the third sub key Z_3.
4. Multiply X_4 and the fourth sub key Z_4.
5. Bitwise XOR the results of steps 1 and 3.
6. Bitwise XOR the results of steps 2 and 4.
7. Multiply the result of step 5 and the fifth sub key Z_5.
8. Add the results of steps 6 and 7.
9. Multiply the result of step 8 and the sixth sub key Z_6.
10. Add the results of steps 7 and 9.
11. Bitwise XOR the results of steps 1 and 9.
12. Bitwise XOR the results of steps 3 and 9.
13. Bitwise XOR the results of steps 2 and 10.
14. Bitwise XOR the results of steps 4 and 10.

For every round except the final transformation, a swap occurs, and the input to the next round is: result of step 11 $\|$ result of step 13 $\|$ result of step 12 $\|$ result of step 14, which becomes $X_1 \| X_2 \| X_3 \| X_4$, the input for the next round. After round (N-1), a N^{th} "half round" final transformation occurs:

1. Multiply X_1 and the first sub key.
2. Add X_2 and the second sub key.
3. Add X_3 and the third sub key.
4. Multiply X_4 and the fourth sub key.

The concatenation of the blocks is the output.

III. KEY SHEDULING

For every round (except final round) we need 6 keys and 4 keys for the final round to perform the encryption process. Thus we can use a mathematical expression for the total no. of keys needed in encryption process.

Total no. of keys needed for (n) rounds = $\{6 * (n-1) + 4\}$.

We use the similar concept that is used in IDEA algorithm. We divide 128 bit key into 16 bit's 8 sub blocks. First 6 keys are used in first round and the remaining two keys are used in next round. Then left shift 128 bit key block by 25 and again we get 8 keys of 16 bits. We repeat the process until we get $\{6*(n-1)+4\}$ keys.

1st round	$Z_1^{(1)}\ Z_2^{(1)}\ Z_3^{(1)}\ Z_4^{(1)}\ Z_5^{(1)}\ Z_6^{(1)}$
2nd round	$Z_1^{(2)}\ Z_2^{(2)}\ Z_3^{(2)}\ Z_4^{(2)}\ Z_5^{(2)}\ Z_6^{(2)}$
3rd round	$Z_1^{(3)}\ Z_2^{(3)}\ Z_3^{(3)}\ Z_4^{(3)}\ Z_5^{(3)}\ Z_6^{(3)}$
. . .	
N-1th round	$Z_1^{(n-1)}\ Z_2^{(n-1)}\ Z_3^{(n-1)}\ Z_4^{(n-1)}\ Z_5^{(n-1)}\ Z_6^{(n-1)}$
Nth round	$Z_1^{(n)}\ Z_2^{(n)}\ Z_3^{(n)}\ Z_4^{(n)}$

Figure 2: Keys used for encryption.

Similarly for decryption we need same no. of keys as used in encryption. The only difference is that for decryption we use multiplicative inverse of the key, which we used for encryption.

IV. DESCRYPTION OF DECRYPTION ALGORITHM

The similarity of encryption and decryption means that decryption is essentially the same process as encryption, the only difference being that different key sub blocks are used. This similarity results from using the output transformation in the encryption process so that the effect of (Z_1; Z_2; Z_3; Z_4) can be cancelled by using inverse key sub blocks (Z_1^{-1};Z_2^{-1};$-Z_3$;$-Z_4$) in the decryption process, as shown in Fig. 3.

1st round	$Z_1^{(n)-1}\ Z_2^{(n)-1}\ -Z_3^{(n)}\ -Z_4^{(n)}\ Z_5^{(n-1)}\ Z_6^{(n-1)}$
2nd round	$Z_1^{(n-1)-1}\ Z_2^{(n-1)-1}\ -Z_3^{(n-1)}\ -Z_4^{(n-1)}\ Z_5^{(n-2)}\ Z_6^{(n-2)}$
3rd round	$Z_1^{(n-2)-1}\ Z_2^{(n-2)-1}\ -Z_3^{(n-2)}\ -Z_4^{(n-2)}\ Z_5^{(n-3)}\ Z_6^{(n-3)}$
. . .	
N-1th round	$Z_1^{(2)-1}\ Z_2^{(2)-1}\ -Z_3^{(2)}\ -Z_4^{(2)}\ Z_5^{(1)}\ Z_6^{(1)}$
Nth round	$Z_1^{(1)-1}\ Z_2^{(1)-1}\ -Z_3^{(1)}\ -Z_4^{(1)}$

Figure 3: Keys used for decryption.

V. RESULTS

We are executing our c program on a turbo c/c++ compiler which is running on an Intel(R) Pentium(R) 4 CPU 3.20 GHz 632 MB of RAM and taking results as total time taken in encryption and decryption of several files. These files are different with each other according to their types and sizes. This time is calculated as min:sec:hund.

Table 1: Table for Encryption Time

No. of Rounds	Total time in encryption (min:sec:hund)		
	Kd01.pdf (1,955 KB)	Kd02.mp3 (5.04 MB)	Kd03.flv (220 MB)
8	00:00:49	00:01:48	01:03:22
9	00:00:54	00:01:54	01:06:46
10	00:00:59	00:01:59	01:08:77

Table 2: Table For Decryption Time

No. of Rounds	Total time in decryption (min:sec:hund)		
	Kd01.pdf (1,955 KB)	Kd02.mp3 (5.04 MB)	Kd03.flv (220 MB)
8	00:00:55	00:01:70	02:30:83
9	00:00:60	00:01:87	02:31:76
10	00:00:65	00:01:95	02:32:86

VI. SUMMARY AND CONCLUSION

In this paper first we introduce the concept of IDEA with variable rounds and its architecture. Then we took results on various files.

This paper shows that execution time increases when the no. of rounds are increased. But it also shows that security will also increased when no. of rounds of IDEA algorithm are unknown to the attacker because he'll have to try for two things now, key as well as no. of rounds.

REFERENCES

[1] X.Lai and James L.Massey, "A Proposal for a New Block Encryption Standard", Advances in Cryptology-EUROCRYPT'90, Springer-Verlag Berlin 1991, pp.389-404 .

[2] Eli Biham, Alex Biryukov, Adi Shamir, Miss in the Middle Attacks on IDEA and Khufu, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 124–138, Springer-Verlag, 1999.

[3] Eli Biham, Orr Dunkelman, Nathan Keller, Related-Key Boomerang and Rectangle Attacks, Advances in Cryptology, proceedings of EUROCRYPT'05, Lecture Notes in Computer Science 3494, pp. 507-525, Springer-Verlag, 2005.

[4] Alex Biryukov, Jorge Nakahara Jr., Bart Preneel, Joos Vandewalle, New Weak Key Classes of IDEA,

proceedings of Information and Communications Security 4, Lecture Notes in Computer Science 2513, pp. 315–326, Springer-Verlag, 2002.

[5] Johan Borst, Lars R. Knudsen, Vincent Rijmen, Two Attacks on Reduced Round IDEA, Advances in Cryptology, proceedings of EUROCRYPT '97, Lecture Notes in Computer Science 1233, pp. 1–13, Springer-Verlag, 1993.

[6] P. Hawkes, L. O'Connor, On Applying Linear Cryptanalysis to IDEA, Advances in Cryptology - Proceedings of ASIACRYPT'96, Lecture Notes in Computer Science 1163, pp. 105–115, Springer-Verlag, 1996.

[7] Pascal Junod, New Attacks Against Reduced-Round Versions of IDEA, proceedings of Fast Software Encryption 12, Lecture Notes in Computer Science 3557, pp. 384–397, Springer-Verlag, 2005.

[8] Willi Meier, On the Security of the IDEA Block Cipher, Advances in Cryptology, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 765, pp. 371–385, Springer-Verlag, 1994.

[9] Jorge Nakahara Jr., Paulo S.L.M. Barreto, Bart Preneel, Joos Vandewalle, Hae Y. Kim, SQUARE Attacks Against Reduced-Round PES and IDEA Block Ciphers, IACR Cryptology ePrint Archive, Report 2001/068, 2001.

[10] Schneier, B. 1996. Applied Cryptography, Second Edition. Wiley.

