Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Impact of Implementation of Cryptographic IDEA Algorithms for Secure Data Communication

Ms Snehal Patil [1], Prof.Archana C.Lomte [2]

Department of Computer Science and Engineering

Bhivarabai Sawant Institute of Technology & Research (BSIOTR), India

Abstract:- Personal privacy is of utmost importance in the global networked world. One of the best tools to help people safeguard their personal information is the use of cryptography. The IDEA algorithm is an interesting one. It consists of a certain number of steps which, primarily, make it seem like having a non-reversible hash function in place of a block cipher. IDEA utilizes 52 sub keys, each of which is 16 bits long. Two bits are employed during each proper round while four are utilized prior to each round and after the completion of the last one. It has a total of eight rounds. The block plain text of IDEA is split into four separate 16 bit-long quarters. Three functions are employed in IDEA to bring together two 16 bit values to create a 16 bit outcome, totaling, XOR and multiplication. For decryption of data encoded with IDEA, the receiving machine's connection needs to identify a matching IDEA key, the key phrase of which can be even 255 characters long. IDEA encryption is considerably quicker and usually thought to be significantly more secure than any DES encryption. Depending on the fastness of a CPU, any permutation of DES, IDEA and key file encryption can be successfully enabled. To measure the power of IDEA against differential crypto analysis, designers have conducted repeated and thorough analyses and the research community has brought it under numerous attacks at the time of rigorous testing. It was found that IDEA is indeed safe and protected under most assumptions. This is because no weak links, be of the algebraic or linear kinds have been found in IDEA. IDEA is one of the most secure and fastest algorithms that the public could possibly use.

Keywords: Cryptographic Algorithm, International data encryption algorithm(IDEA) etc.

I. Introduction:-

The Data Encryption Standard (DES) algorithm has been a popular secret key encryption algorithm and is used in many commercial and financial applications. The block cipher IDEA operates with 64-bit plaintext and cipher text blocks and is controlled by a 128-bit key. The fundamental innovation in the design of this algorithm is the use of operations from three different algebraic groups. The substitution boxes and the associated table lookups used in the block ciphers available to-date have been completely avoided. The algorithm structure has been chosen such that, with the exception that different key sub-blocks are used, the encryption process is identical to the decryption process. IDEA is a patented and universally applicable block encryption algorithm, which permits the effective protection of transmitted and stored data against unauthorized access by third parties. With a key of 128 bits in length, IDEA is far more secure than the widely known DES based on a 56-bit key. The fundamental criteria for the development of IDEA were military strength for all security requirements and easy hardware and software implementation. The algorithm is used worldwide in various banking and industry applications. They predestine the algorithm for use in a great number of commercial applications.

II. Key Generation

The 64-bit plaintext block is partitioned into four 16-bit sub-blocks, since all the algebraic operations used in the encryption process operate on 16-bit numbers. Another process produces for each of the encryption rounds, six 16-bit key sub-blocks from the 128-bit key. Since a further four 16-bit key-sub-blocks are required for the subsequent output transformation, a total of 52 (= 8 x 6 + 4) different 16-bit sub-blocks have to be generated from the 128-bit key. The key sub-blocks used for the encryption and the decryption in the individual rounds are shown in Table 1.

Round 1	$Z_1^{(1)}\ Z_2^{(1)}\ Z_3^{(1)}\ Z_4^{(1)}\ Z_5^{(1)}\ Z_6^{(1)}$
Round 2	$Z_1^{(2)}\ Z_2^{(2)}\ Z_3^{(2)}\ Z_4^{(2)}\ Z_5^{(2)}\ Z_6^{(2)}$
Round 3	$Z_1^{(3)}\ Z_2^{(3)}\ Z_3^{(3)}\ Z_4^{(3)}\ Z_5^{(3)}\ Z_6^{(3)}$
Round 4	$Z_1^{(4)}\ Z_2^{(4)}\ Z_3^{(4)}\ Z_4^{(4)}\ Z_5^{(4)}\ Z_6^{(4)}$
Round 5	$Z_1^{(5)}\ Z_2^{(5)}\ Z_3^{(5)}\ Z_4^{(5)}\ Z_5^{(5)}\ Z_6^{(5)}$
Round 6	$Z_1^{(6)}\ Z_2^{(6)}\ Z_3^{(6)}\ Z_4^{(6)}\ Z_5^{(6)}\ Z_6^{(6)}$
Round 7	$Z_1^{(7)}\ Z_2^{(7)}\ Z_3^{(7)}\ Z_4^{(7)}\ Z_5^{(7)}\ Z_6^{(7)}$
Round 8	$Z_1^{(8)}\ Z_2^{(8)}\ Z_3^{(8)}\ Z_4^{(8)}\ Z_5^{(8)}\ Z_6^{(8)}$
Output Transform	$Z_1^{(9)}\ Z_2^{(9)}\ Z_3^{(9)}\ Z_4^{(9)}$

© 2013, IJARCSSE All Rights Reserved

The 52 16-bit key sub-blocks which are generated from the 128-bit key are produced as follows:

- First, the 128-bit key is partitioned into eight 16-bit sub-blocks which are then directly used as the first eight key sub-blocks.
- The 128-bit key is then cyclically shifted to the left by 25 positions, after which the resulting 128-bit block is again partitioned into eight 16-bit sub-blocks to be directly used as the next eight key sub-blocks.
- The cyclic shift procedure described above is repeated until all of the required 52 16-bit key sub-blocks have been generated.

The IDEA algorithm is interesting in its own right. It includes some steps which, at first, make it appear that it might be a non-invertible hash function instead of a block cipher. Also, it is interesting in that it entirely avoids the use of any lookup tables or S-boxes. IDEA uses 52 subkeys, each 16 bits long. Two are used during each round proper, and four are used before every round and after the last round. It has eight rounds. The plaintext block in IDEA is divided into four quarters, each 16 bits long. Three operations are used in IDEA to combine two 16 bit values to produce a 16 bit result, addition, XOR, and multiplication. Addition is normal addition with carries, modulo 65,536. Multiplication, as used in IDEA, requires some explanation. Multiplication by zero always produces zero, and is not invertible. Multiplication modulo n is also not invertible whenever it is by a number which is not relatively prime to n. The way multiplication is used in IDEA, it is necessary that it be always invertible. This is true of multiplication IDEA style.

The number 65,537, which is 2^16+1, is a prime number. (Incidentally, 2^8+1, or 257, is also prime, and so is 2^4+1, or 17, but 2^32+1 is not prime, so IDEA cannot be trivially scaled up to a 128-bit block size.) Thus, if one forms a multiplication table for the numbers from 1 through 65,536, each row and column will contain every number once only, forming a Latin square, and providing an invertible operation. The numbers that 16 bits normally represent are from 0 to 65,535 (or, perhaps even more commonly, from -32,768 to 32,767). In IDEA, for purposes of multiplication, a 16 bit word containing all zeroes is considered to represent the number 65,536 other numbers are represented in conventional unsigned notation, and multiplication is modulo the prime number 65,537.

The IDEA encryption algorithm

- provides high level security not based on keeping the algorithm a secret, but rather upon ignorance of the secret key
- is fully specified and easily understood
- is available to everybody
- is suitable for use in a wide range of applications
- can be economically implemented in electronic components (VLSI Chip)
- can be used efficiently
- may be exported world wide
- is patent protected to prevent fraud and piracy.

Flow of IDEA(Algorithm)

Four quarters of the plaintext be called A, B, C, and D, and the 52 subkeys called K(1) through K(52). Before round 1, or as the first part of it, the following is done:

Multiply A by K(1). Add K(2) to B. Add K(3) to C. Multiply D by K(4).

Round 1 proper consists of the following:

Calculate A xor C (call it E) and B xor D (call it F).

Multiply E by K(5). Add the new value of E to F.

Multiply the new value of F by K(6). Add the result, which is also the new value of F, to E.

Change both A and C by XORing the current value of F with each of them change both B and D by XORing the current value of E with each of them.

Swap B and C.

Repeat all of this eight times, or seven more times, using K(7) through K(12) the second time, up to K(43) through K(48) the eighth time. Note that the swap of B and C is *not* performed after round 8.

Then multiply A by K(49). Add K(50) to B. Add K(51) to C. Multiply D by K(52).

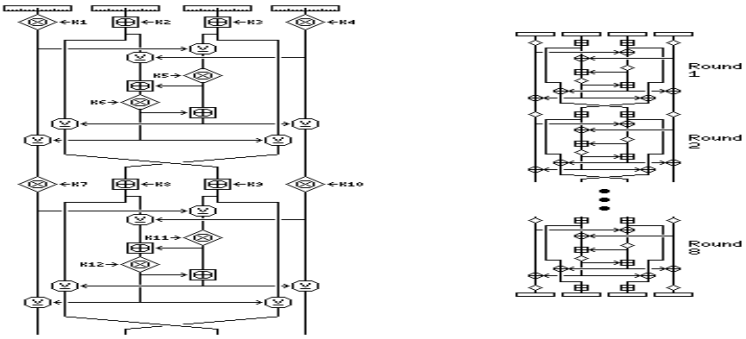

Fig 1 IDEA Encryption Process

© 2013, IJARCSSE All Rights Reserved

III. Decryption

How can the round in IDEA be reversed, since all four quarters of the block are changed at the same time, based on a function of all four of their old values? Well, the trick to that is that A xor C isn't changed when both A and C are XORed by the same value, that value cancels out, no matter what that value might be. And the same applies to B xor D. And since the values used are functions of (A xor C) and (B xor D), they are still available.

This cross-footed round, rather than a Feistel round, is the most striking distinguishing factor of IDEA, although its use of multiplication, addition, and XOR to avoid the use of S-boxes is also important.

Those that are added are replaced by their two's complement. Those that are multiplied in are replaced by their multiplicative inverse, modulo 65,537, in IDEA notation when used to change blocks directly, but those used to calculate the cross-footed F-functions are not changed. Keys XORed in would not need to be changed, but there aren't any such keys in IDEA. Due to the placement of the swap, the first four keys for decryption are moved somewhat differently than the other keys used for the same operation between rounds.

The decryption key schedule is:

The first four subkeys for decryption are:

KD(1) = 1/K(49)
KD(2) = -K(50)
KD(3) = -K(51)
KD(4) = 1/K(52)

and they do not quite follow the same pattern as the remaining subkeys which follow.

The following is repeated eight times, adding 6 to every decryption key's index and subtracting 6 from every encryption key's index:

KD(5) = K(47)
KD(6) = K(48)
KD(7) = 1/K(43)
KD(8) = -K(45)
KD(9) = -K(44)
KD(10) = 1/K(46)

Subkey generation

The 128-bit key of IDEA is taken as the first eight subkeys, K(1) through K(8). The next eight subkeys are obtained the same way, after a 25-bit circular left shift, and this is repeated until all encryption subkeys are derived.

Applications

Today, there are hundreds of IDEA-based security solutions available in many market areas, ranging from Financial Services, and Broadcasting to Government. IDEA is the name of a proven, secure, and universally applicable block encryption algorithm, which permits effective protection of transmitted and stored data against unauthorized access by third parties. The fundamental criteria for the development of IDEA were highest security requirements along with easy hardware and software implementation for fast execution.

- The IDEA algorithm can easily be embedded in any encryption software.
- Data encryption can be used to protect data transmission and storage. Typical fields are Audio and video data for cable TV, pay TV, video conferencing, distance learning.
- Sensitive financial and commercial data Email via public networks Transmission links via modem, router or ATM link, GSM technology.
- Smart cards

IV. Conclusion:-

The basic aim of this paper is to increase the strength of existing IDEA algorithm. The proposed algorithm International data Encryption algorithm (S-IDEA) has two key features:-increased key size (256 bits) and increased degree of diffusion (two MA blocks are used in a single round instead of one). The 104 sub-keys are being used as compared to 52 sub-keys previously which enhance the complexity of confusion. Therefore the probability of other forms of attack is reduced due to amount of work that has to be carried out when 104 sub-keys are involved. Addition of a new MA block in each round of SIDEA has contributed to an increase in complexity of diffusion. It makes the algorithm more secure and less susceptible to cryptanalysis.

References

[1] R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber, and W. Fichtner, “A 177Mb/sec VLSI implementation of the international data encryption algorithm," IEEE Journal of Solid-State Circuits, vol. 29, pp. 303-307, March 1994.

[2] S. Wolter, H. Matz, A. Schubert, and R. Laur, “On the VLSI implementation of the international data encryption algorithm IDEA," in Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 1, pp. 397-400, 1995.

[3] Harivans Pratap singh “secure International Data Encryption” International Journal of Advance Research in Electrical and Instrumentation Engineering Vol 2 issue 2 ,February 2013

[4] A.D Chaudhari “Implementation of IDEA Architecture “ International journal of Innovative Technology and Exploring Enginerring ISSN:2278-3075 vol-3 issue 1,june 2013

© 2013, IJARCSSE All Rights Reserved