JOJAPS

eISSN 2504-8457

KEMENTERIAN PENDIDIKAN TINGGI

POLITEKNIK MALAYSIA PORT DICKSON

Journal Online Jaringan COT POLIPD (JOJAPS)

Data Security with International Data Encryption Algorithm

Robbi Rahim[1*], M Mesran[2], Muhammad Syahrizal[2], Andysah Putera Utama Siahaan[3]

[1,3]School of Computer and Communication Engineering, Universiti Malaysia Perlis, Malaysia
[2]Department of Informatics Engineering, STMIK Budidarma, Medan, Indonesia
**usurobbi85@zoho.com*

Abstract

Information security is very important nowadays, how to secure data certainly requires a technique one of which is the use of cryptography, IDEA algorithm is one of cryptographic algorithms that can be used to secure the message, and in this research IDEA algorithm process is displayed gradually to facilitate the development of IDEA algorithm in various purposes

Keywords— Encryption Decryption, IDEA Algorithm, Secure Message

INTRODUCTION

Cryptographic methods are used to ensure confidential data to be unknown to others [1] [2] [3] [4] [5]. Cryptographic methods can be used to secure various types of data by using certain algorithms, and each cryptographic algorithm has its advantages and disadvantages [4] [5]. However, the biggest problem is how to know and understand the workings of the cryptographic algorithm so that the algorithm can be used optimally for the desired object [4], one of the cryptographic algorithms that could be utilized is the International Data Encryption Algorithm (IDEA).

The IDEA algorithm uses confusion and diffusion in its encryption process. Different from other block cipher methods, IDEA uses incompatible algebraic operations XOR, module 216 addition, and multiplication modulo 216 + 1. This multiplex operation modulo 216 + 1 replaces the Substitution Box (S-Box) [6] [7]. In this paper will explain the working procedures of IDEA algorithm in securing messages both SMS Messages, Text Messages, Word and so on in the form of text, It is expected that with the publication of the working procedure IDEA algorithm can be implemented in stages for various purposes of data security, besides how the workings of IDEA algorithm in the key generation, process of encryption and decryption is also very important and in this paper is displayed step by step so the reader know how the process working.

METHODOLOGY

A. Cryptography

Cryptography is a field of science that learns about how to conceal an important information into a form that cannot be read by anyone and return it back to the original data by using various techniques that already exist so that the information cannot be known by any party who is not the owner or who are not interested [3] [4] [8]. The other side of cryptography is cryptanalysis which is the study of how to solve cryptographic mechanisms.

For most people, cryptography takes precedence in keeping communication secret and extraordinary. As has been known and agreed that the protection of sensitive communication has been of particular concern to the importance of using cryptography. However, this is only part of the application of cryptography today [4] [9].

B. International Data Encryption Algorithm

IDEA is a block cipher algorithm that operates on a 64-bit plaintext block. The key length is 128 bits, by encryption and decryption using the same (symmetrical) key [6], the IDEA algorithm uses confusion and diffusion on encryption and uses the following incompatible algebraic operations [6] [7]:

1. XOR.
2. Added modulo 216.
3. Multiplication modulo 216 + 1 (this operation replaces the S-box or S-Box).

The IDEA algorithm uses multiplication modulo 216 + 1 with the consideration that multiplication with zero always yields zero and has no inversion. Multiplication modulo n also has no inversion if the number multiplied is not relatively prime to *n*. While cryptographic algorithms require operations that have inversions. The number 65537 (216 + 1) is a prime number. Therefore, modulo multiplication operation (216 + 1) on the IDEA algorithm has an inversion, if forming a multiplication table for numbers ranging from 1 to 65536, each row and column contains only one number once. In IDEA, for multiplication operations, a 16-bit number consisting of zeros is all considered a number 65536, while other numbers remain under the unmarked numbers it represents. This IDEA algorithm could be divided into three parts, such as key generation, encryption and decryption [6] [7] [10] [11].

C. Key Generation

The process of key generation begins by dividing 128-bit keys into eight pieces of the 16-bit subkey. These are the first eight subkeys for the algorithm with details of the first six subkeys for round 1 and the last two subkeys for round 2. The key is rotated 25 bits to the left and is divided into eight subkeys again. These are the eight-second subkeys for the algorithm with the details of the first four subkeys for round 2 and the last four for the round 3. The algorithm uses only 52 subkeys with six subkeys for eight rounds plus four subkeys for output transformation [10] [11], see figure below:

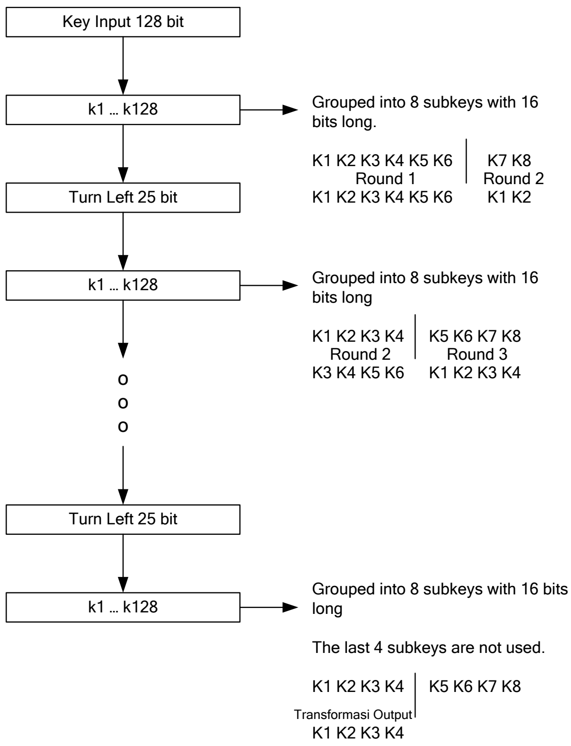

Fig 1. Key Generation Diagram

D. Encryption Process

The IDEA algorithm encryption process is as follows:64-bit plaintext is split into four sub-blocks with 16 bits long, i.e., X1, X2, X3, X4. These four sub-blocks serve as input for the first-phase iteration of the algorithm. There is a total of 8 iterations. At each iteration, four sub-blocks are XOR-aligned, added, multiplied by the other and with six 16-bit subkeys. Among the iterations of the second and third sub-blocks are interchangeable. Finally, 4 sub-blocks are merged with four subkeys in the output transformation [6] [10] [11], The IDEA algorithm encryption process could be seen in the following figure:

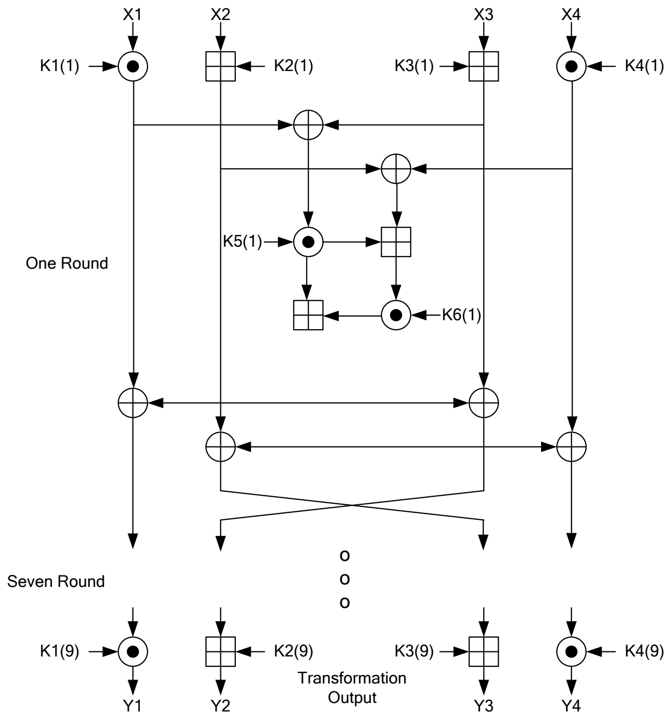

Fig 2. Encryption Diagram

E. Decryption Process

The decryption process is the same as the encryption process. The only difference lies in the rules of the subkey. The subkey order is inverted with the encryption process and its subkey is inverse. The subkey on the output transformation step of the encryption process is invoked and used as a subkey in round 1 of the decryption process. Subkeys in round 8 are inverse and used as subkey on round 1 and 2 in the decryption process [10] [11].

RESULT AND DISCUSSION

For the process of encryption and decryption testing with IDEA algorithm, the first step is to determine the key first
Key = ConferenceMalang

The key generation process as below:

1. First Round

Key in hexadecimal: 436F6E666572656E63654D616C616E67
Split into 8 parts
KE1 (Round 1) = 436F
KE2 (Round 1) = 6E66
KE3 (Round 1) = 6572
KE4 (Round 1) = 656E
KE5 (Round 1) = 6365
KE6 (Round 1) = 4D61
KE1 (Round 2) = 6C61
KE2 (Round 2) = 6E67

2. Second Round

Rotate Left (436F6E666572656E63654D616C616E67, 25) = CCCAE4CADCC6CA9AC2D8C2DCCE86DEDC
Split into 8 parts:
KE3 (Round 2) = CCCA
KE4 (Round 2) = E4CA
KE5 (Round 2) = DCC6
KE6 (Round 2) = CA9A
KE1 (Round 3) = C2D8

KE2 (Round 3) = C2DC
KE3 (Round 3) = CE86
KE4 (Round 3) = DEDC

3. Third Round
RotateLeft (CCCAE4CADCC6CA9AC2D8C2DCCE86DEDC, 25) = 95B98D953585B185B99D0DBDB99995C9

Split into 8 parts
KE5 (Round 3) = 95B9
KE6 (Round 3) = 8D95
KE1 (Round 4) = 3585
KE2 (Round 4) = B185
KE3 (Round 4) = B99D
KE4 (Round 4) = 0DBD
KE5 (Round 4) = B999
KE6 (Round 4) = 95C9

Do the rotation until the 8th round so that the results obtained are as follows:

Rotate Left
(E86DEDCCCCAE4CADCC6CA9AC2D8C2DCC, 25) = 99995C995B98D953585B185B99D0DBDB

Split into 8 parts (last 4 parts are not used):
KE1 (Transformation Output) = 9999
KE2 (Transformation Output) = 5C99
KE3 (Transformation Output) = 5B98
KE4 (Transformation Output) = D953

After determining the key used, the next is to encrypt the message process, below are the process:
Message = ICEEIEOK
Key = ConferenceMalang

1. First Round
01) L#1 = (X1 * K1) mod (2^16 + 1) = 4943 * 436F mod (2^16 + 1) = 39C1
02) L#2 = (X2 + K2) mod 2^16 = 4545 + 6E66 mod 2^16 = B3AB
03) L#3 = (X3 + K3) mod 2^16 = 4945 + 6572 mod 2^16 = AEB7
04) L#4 = (X4 * K4) mod (2^16 + 1) = 4F4B * 656E mod (2^16 + 1) = 89D0
05) L#5 = L#1 XOR L#3 = 39C1 XOR AEB7 = 9776
06) L#6 = L#2 XOR L#4 = B3AB XOR 89D0 = 3A7B
07) L#7 = (L#5 * K5) mod (2^16 + 1) = 9776 * 6365 mod (2^16 + 1) = 28C0
08) L#8 = (L#6 + L#7)) mod 2^16 = 3A7B + 28C0 mod 2^16 = 633B
09) L#9 = (L#8 * K6) mod (2^16 + 1) = 633B * 4D61 mod (2^16 + 1) = 3A5D
10) L#10 = (L#7 + L#9)) mod 2^16 = 28C0 + 3A5D mod 2^16 = 631D
11) L#11 = L#1 XOR L#9 = 39C1 XOR 3A5D = 039C
12) L#12 = L#3 XOR L#9 = AEB7 XOR 3A5D = 94EA
13) L#13 = L#2 XOR L#10 = B3AB XOR 631D = D0B6
14) L#14 = L#4 XOR L#10 = 89D0 XOR 631D = EACD

For the next round the round key is used:
X1 = L#11 = 039C
X2 = L#12 = 94EA
X3 = L#13 = D0B6
X4 = L#14 = EACD

2. Second Round
01) L#1 = (X1 * K1) mod (2^16 + 1) = 039C * 6C61 mod (2^16 + 1) = 2C95
02) L#2 = (X2 + K2) mod 2^16 = 94EA + 6E67 mod 2^16 = 0351

03) L#3 = (X3 + K3) mod 2^16 = D0B6 + CCCA mod 2^16 = 9D80
04) L#4 = (X4 * K4) mod (2^16 + 1) = EACD * E4CA mod (2^16 + 1) = 07EB
05) L#5 = L#1 XOR L#3 = 2C95 XOR 9D80 = B115
06) L#6 = L#2 XOR L#4 = 0351 XOR 07EB = 04BA
07) L#7 = (L#5 * K5) mod (2^16 + 1) = B115 * DCC6 mod (2^16 + 1) = 6988
08) L#8 = (L#6 + L#7)) mod 2^16 = 04BA + 6988 mod 2^16 = 6E42
09) L#9 = (L#8 * K6) mod (2^16 + 1) = 6E42 * CA9A mod (2^16 + 1) = 1072
10) L#10 = (L#7 + L#9)) mod 2^16 = 6988 + 1072 mod 2^16 = 79FA
11) L#11 = L#1 XOR L#9 = 2C95 XOR 1072 = 3CE7
12) L#12 = L#3 XOR L#9 = 9D80 XOR 1072 = 8DF2
13) L#13 = L#2 XOR L#10 = 0351 XOR 79FA = 7AAB
14) L#14 = L#4 XOR L#10 = 07EB XOR 79FA = 7E11

For the next round the round key is used:
X1 = L#11 = 3CE7
X2 = L#12 = 8DF2
X3 = L#13 = 7AAB
X4 = L#14 = 7E11

3. Third Round
01) L#1 = (X1 * K1) mod (2^16 + 1) = 3CE7 * C2D8 mod (2^16 + 1) = 428E
02) L#2 = (X2 + K2) mod 2^16 = 8DF2 + C2DC mod 2^16 = 50CE
03) L#3 = (X3 + K3) mod 2^16 = 7AAB + CE86 mod 2^16 = 4931
04) L#4 = (X4 * K4) mod (2^16 + 1) = 7E11 * DEDC mod (2^16 + 1) = A6DE
05) L#5 = L#1 XOR L#3 = 428E XOR 4931 = 0BBF
06) L#6 = L#2 XOR L#4 = 50CE XOR A6DE = F610
07) L#7 = (L#5 * K5) mod (2^16 + 1) = 0BBF * 95B9 mod (2^16 + 1) = A129
08) L#8 = (L#6 + L#7)) mod 2^16 = F610 + A129 mod 2^16 = 9739
09) L#9 = (L#8 * K6) mod (2^16 + 1) = 9739 * 8D95 mod (2^16 + 1) = 158B
10) L#10 = (L#7 + L#9)) mod 2^16 = A129 + 158B mod 2^16 = B6B4
11) L#11 = L#1 XOR L#9 = 428E XOR 158B = 5705
12) L#12 = L#3 XOR L#9 = 4931 XOR 158B = 5CBA
13) L#13 = L#2 XOR L#10 = 50CE XOR B6B4 = E67A
14) L#14 = L#4 XOR L#10 = A6DE XOR B6B4 = 106A

For the next round the round key is used:
X1 = L#11 = 5705
X2 = L#12 = 5CBA
X3 = L#13 = E67A
X4 = L#14 = 106A

The rotation process is performed until the 8th round with the following results:

01) L#1 = (X1 * K1) mod (2^16 + 1) = 4F3A * CCAE mod (2^16 + 1) = D215
02) L#2 = (X2 + K2) mod 2^16 = EC8A + 4CAD mod 2^16 = 3937
03) L#3 = (X3 + K3) mod 2^16 = 6BFF + CC6C mod 2^16 = 386B
04) L#4 = (X4 * K4) mod (2^16 + 1) = 26DC * A9AC mod (2^16 + 1) = 3E0F
05) L#5 = L#1 XOR L#3 = D215 XOR 386B = EA7E
06) L#6 = L#2 XOR L#4 = 3937 XOR 3E0F = 0738
07) L#7 = (L#5 * K5) mod (2^16 + 1) = EA7E * 2D8C mod (2^16 + 1) = 3930
08) L#8 = (L#6 + L#7)) mod 2^16 = 0738 + 3930 mod 2^16 = 4068
09) L#9 = (L#8 * K6) mod (2^16 + 1) = 4068 * 2DCC mod (2^16 + 1) = 8F5B
10) L#10 = (L#7 + L#9)) mod 2^16 = 3930 + 8F5B mod 2^16 = C88B
11) L#11 = L#1 XOR L#9 = D215 XOR 8F5B = 5D4E
12) L#12 = L#3 XOR L#9 = 386B XOR 8F5B = B730
13) L#13 = L#2 XOR L#10 = 3937 XOR C88B = F1BC
14) L#14 = L#4 XOR L#10 = 3E0F XOR C88B = F684

For transformation output
X1 = L#11 = 5D4E
X2 = L#13 = F1BC
X3 = L#12 = B730
X4 = L#14 = F684

Transformation output process
01) Y1 = (X1 * K1) mod (2^16 + 1) = 5D4E * 9999 mod (2^16 + 1) = 29A3
02) Y2 = (X2 + K2) mod 2^16 = F1BC + 5C99 mod 2^16 = 4E55

03) Y3 = (X3 + K3) mod 2^16 = B730 + 5B98 mod 2^16 = 12C8
04) Y4 = (X4 * K4) mod (2^16 + 1) = F684 * D953 mod (2^16 + 1) = FF88

Ciphertext Results:
Y1 = 29A3 =)£
Y2 = 4E55 = NU
Y3 = 12C8 = È
Y4 = FF88 = ÿˆ
Ciphertext =)£NUÈÿˆ

The encryption process successfully applied with the results as above, for the operation of decrypting just do the inverse process that already described.

CONCLUSION

Message security process by using IDEA algorithm successfully done, where the work process of the IDEA algorithm is displayed gradually so that for implementation into the application form is applied maximally.

REFERENCES

[1] R. Rahim, "128 Bit Hash of Variable Length in Short Message Service Security," *International Journal of Security and Its Applications,* vol. 11, no. 1, pp. 45-58, 2017.

[2] R. Rahim and A. Ikhwan, "Cryptography Technique with Modular Multiplication Block Cipher and Playfair Cipher," *IJSRST,* vol. II, no. 6, pp. 71-78, 2016.

[3] R. Rahim and A. Ikhwan, "Study of Three-Pass Protocol on Data Security," *International Journal of Science and Research (IJSR),* vol. 5, no. 11, pp. 102-104, 2016.

[4] Legito and R. Rahim, "SMS Encryption Using Word Auto Key Encryption," *International Journal of Recent Trends in Engineering & Research (IJRTER),* vol. 3, no. 1, pp. 251-256, 2017.

[5] D. Nofriansyah and R. Rahim, "Combination of Pixel Value Differencing Algorithm with Caesar Cipher Algorithm for Steganography," *International Journal of Research In Science & Engineering,* vol. 2, no. 6, pp. 153-159, 2016.

[6] E. Hariyanto and R. Rahim, "Arnold's Cat Map Algorithm in Digital Image Encryption," *International Journal of Science and Research (IJSR),* vol. 5, no. 10, pp. 1363-1365, 2016.

[7] A. P. U. Siahaan and R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm," *International Journal of Security and its Applications,* vol. 10, no. 8, pp. 173-180, 2016.

[8] H. P. Singh, S. Verma, and S. Mishra, "Secure-International Data Encryption Algorithm," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 2, no. 2, pp. 780-792, 2013.

[9] J. Chen, D. Xue, and X. Lai, "An analysis of international data encryption algorithm(IDEA) security against differential cryptanalysis," *Journal of Natural Sciences,* vol. 13, no. 6, p. 697–701, 2008.

[10] S. Artheeswari and R. Chandrasekaran, "INTERNATIONAL DATA ENCRYPTION ALGORITHM (IDEA) FOR DATA SECURITY IN CLOUD," *International Journal of Technology and Engineering,* vol. 8, no. 1, pp. 6-11, 2016.

[11] O. Almasri and H. M. Jani, "Introducing an Encryption Algorithm based on IDEA," *International Journal of Science and Research (IJSR),* vol. 2, no. 9, pp. 334-339, 2013.