ANALISA FUNGSI HASH DALAM ENKRIPSI IDEA UNTUK KEAMANAN RECORD INFORMASI

Ramen Antonov Purba

Manajemen Informatika, Politeknik Unggul LP3M Medan
Jl.Iskandar Muda No.3 EF Medan-Sumatera Utara, HP : +62 821 6299 0006
www.politeknikunggul.ac.id, info@politeknikunggul.ac.id, ramen_purba@yahoo.com

ABSTRACT

Issues of security and confidentiality of data is very important to organization or individual. If the data in a network of computers connected with a public network such as the Internet. Of course a very important data is viewed or hijacked by unauthorized persons. Because if this happens we will probably corrupted data can be lost even that will cause huge material losses. This research discusses the security system of sending messages/data using the encryption aims to maintain access of security a message from the people who are not authorized/ eligible. Because of this delivery system is very extensive security with the scope then this section is limited only parsing the IDEA Algorithm with hash functions, which include encryption, decryption. By combining the encryption IDEA methods (International Data Encryption Algorithm) to encrypt the contents of the messages/data with the hash function to detect changes the content of messages/data is expected security level to be better. Results from this study a software that can perform encryption and decryption of messages/data, generate the security key based on the message/data is encrypted.

Keywords: IDEA Encryption, Hash Function, Data Security

ABSTRAK

Masalah keamanan dan kerahasiaan data merupakan hal yang sangat penting dalam suatu organisasi maupun pribadi. Apalagi kalau data tersebut berada dalam suatu jaringan komputer yang terhubung/terkoneksi dengan jaringan publik misalnya internet. Tentu saja data yang sangat penting tersebut dilihat atau dibajak oleh orang yang tidak berwenang. Sebab jika hal ini sampai terjadi kemungkinan data kita akan rusak bahkan bisa hilang yang akan menimbulkan kerugian material yang besar. Pada tulisan ini akan dibahas sistem keamanan pengiriman pesan/data dengan menggunakan penyandian yang bertujuan untuk menjaga karahasiaan suatu pesan dari akses orang-orang yang tidak berwenang/ berhak. Karena sistem keamanan pengiriman ini sangat luas cakupannya maka pada bagian ini dibatasi hanya menguraikan Kriptografi menggunakan Algoritma IDEA dan fungsi hash yang meliputi proses enkripsi, dekripsi. International Data Encryption Algorithm (IDEA) adalah metode enkripsi yang dapat diandalkan diantara metode-metode lain yang ada saat ini. Kriptografi IDEA menggunakan key 128-bit dan block plaintext 64-bit sehingga record yang dienkripsi cukup aman.Dengan menggabungkan metode enkripsi IDEA (International Data Encryption Algorithm) untuk mengenkripsi isi pesan/data dan fungsi hash untuk mengetahui adanya perubahan terhadap isi pesan/data diharapkan tingkat keamanan menjadi lebih baik. Hasil dari penelitian ini berupa sebuah perangkat lunak yang mampu melakukan enkripsi dan dekripsi terhadap pesan/data, men-generate security key berdasarkan pesan/data yang dienkripsi.

Kata kunci: Enkripsi IDEA, Fungsi Hash, Keamanan Data

PENDAHULUAN

Pertukaran dan *sharing* informasi pada saat sekarang ini lebih sering dilakukan, sehingga dibutuhkan suatu sistem keamanan yang baik untuk mencegah kejahatan dan serangan-serangan dari pihak lain, khususnya terhadap informasi penting yang hanya boleh diketahui dan digunakan pihak-pihak tertentu saja. Keamanan merupakan salah satu aspek yang penting dalam sebuah sistem. Banyak orang mensiasati bagaimana cara mengamankan informasi yang dikomunikasikan atau bagaimana cara mendeteksi keaslian dari informasi yang diterimanya.

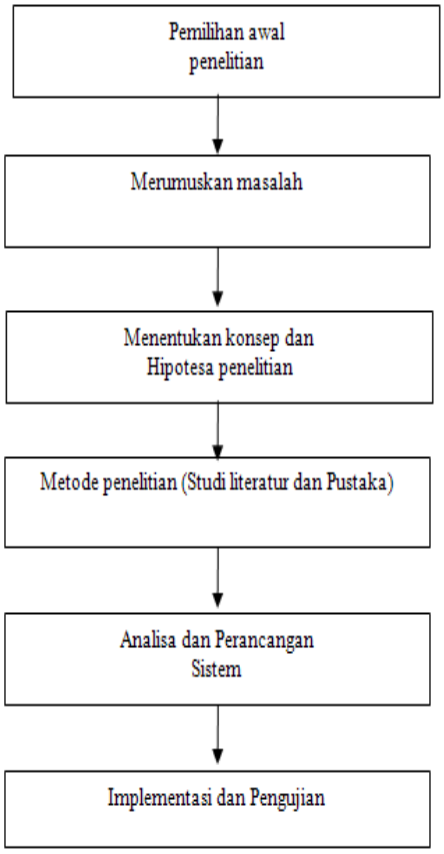
Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (plaintext) ketika pesan dikirim dari suatu tempat ke tempat lain [1]. Tugas utama kriptografi adalah untuk menjaga pesan atau kunci tetap terjaga kerahasiaannya dari penyadap (attacker). Penyadap pesan diasumsikan mempunyai akses yang lengkap dalam saluran komunikasi antara pengirim pesan dan penerima pesan. International Data Encryption Algorithm (IDEA) adalah metode enkripsi yang dapat diandalkan diantara metode-metode lain yang ada saat ini. Kriptografi IDEA menggunakan key 128-bit dan block plaintext 64-bit [2] sehingga record yang dienkripsi cukup aman.

Sedangkan fungsi hash dipakai untuk mengetahui apakah isi suatu record telah diubah atau tidak. Fungsi *hash* merupakan salah satu metode keamanan yang didesain oleh Ronald Rivest (salah satu penemu dari Algoritma RSA) pada tahun 1991. Dalam Kriptografi, fungsi *hash* digunakan secara luas dengan *hash* value 128-bit. Dengan penggunaan metode IDEA dan fungsi *hash* secara bersamaan untuk membangun sebuah sistem keamanan *record*, diharapkan tingkat *Confidentiality* dan *Integrity record* akan lebih baik.

METODE PENELITIAN

Kerangka Kerja

Penelitian yang dilakukan dapat digambarkan dalam suatu alur kegiatan kerja penelitian seperti terlihat pada gambar.



Gambar 1. Kerangka Kerja Penelitian

Uraian Kerja Penelitian

A. Pemilihan Awal Penelitian

Langkah awal yang dilakukan dalam penelitian ini adalah mempelajari dan menentukan masalah yang akan diteliti.

B. Merumuskan Masalah

Dari permasalahan yang ada kemudian dirumuskan sehingga penelitian dapat lebih terarah. Kemudian melanjutkan ke tahap-tahap berikutnya yaitu menentukan judul penelitian.

C. Hipotesis Penelitian

Hipotesis yang akan digunakan dalam penelitian ini adalah, bahwa masalah keamanan merupakan salah satu aspek penting dari sebuah sistem informasi. Proses pertukaran dan *sharing* informasi lebih sering dilakukan, sehingga dibutuhkan suatu sistem keamanan yang baik untuk mencegah serangan-serangan dari pihak lain, khususnya terhadap informasi penting yang hanya boleh diketahui dan digunakan pihak-pihak tertentu saja.

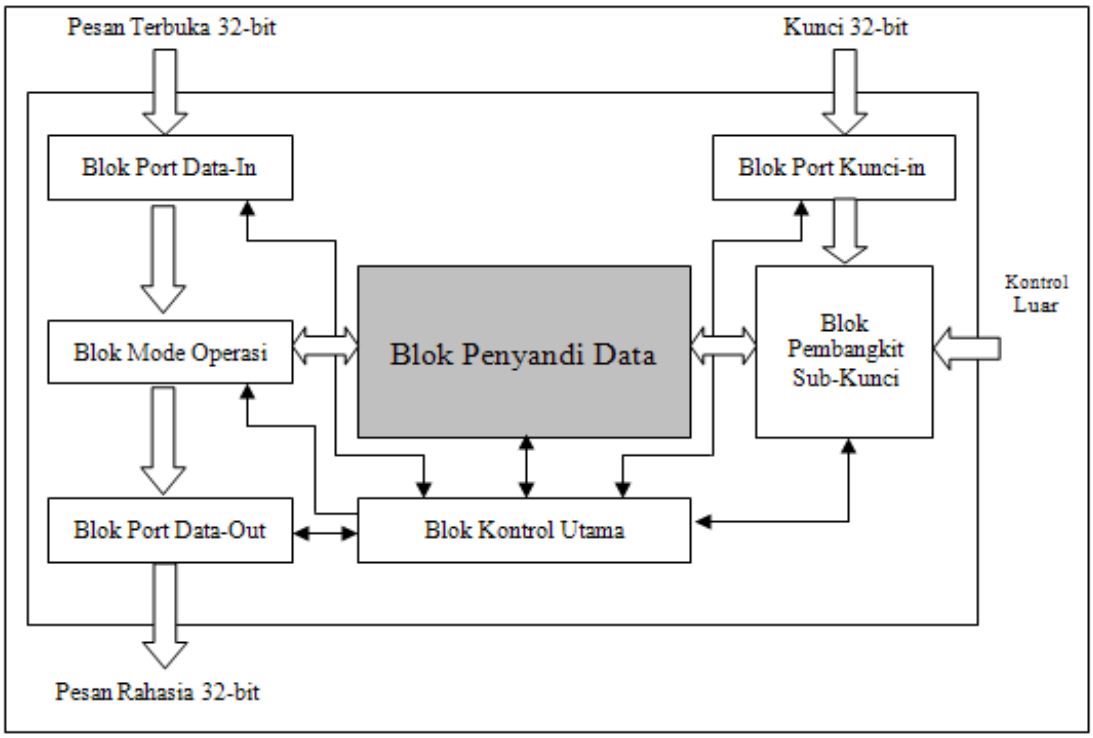
D. Metode Penelitian

Pendekatan yang akan digunakan adalah dengan *survey literature, paper* dan buku-buku yang ada kaitannya dengan judul penelitian. Studi literatur dilakukan untuk mengetahui sumber-sumber dan penyebab terjadinya serangan terhadap komputer atau informasi dan untuk mengetahui bagaimana teknik-teknik dalam mengamankan data.

HASIL DAN PEMBAHASAN

Arsitektur Kiptografi IDEA

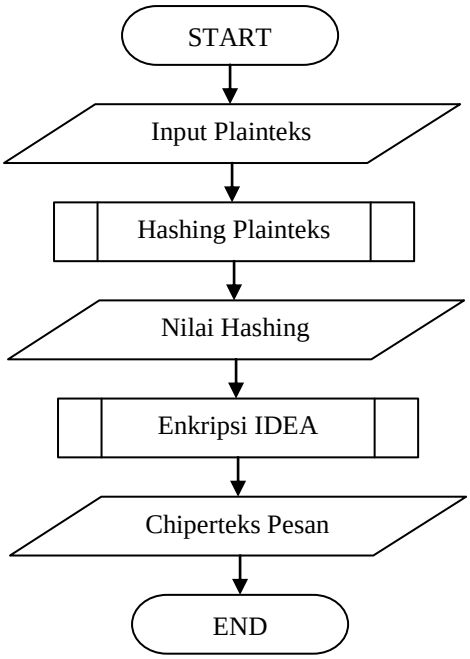
Pada berikut diperlihatkan arsitektur atau gambaran umum sebuah processor yang mengolah sistem keamanan data dengan menggunakan algoritma IDEA.



Gambar 2. Arsitektur IDEA

Analisa Enkripsi

Sistem enkripsi merupakan suatu untuk mengolah data awal (plainteks) menjadi sebuah data acak (chipherteks) yang tidak dapat diterjemahkan secara langsung. Dalam sistem enkripsi ini terdapat proses *hashing*, proses enkripsi IDEA. Proses kerja sistem enkripsi dapat digambarkan menjadi seperti *flowchart* pada gambar.



Gambar 3. *Flowchart* Sistem Enkripsi

Analisa Input Plainteks Pada Enkripsi

Input berupa karakter, dalam pengujian input plainteks 8 karakter. Input ini dapat diproses apabila telah diberikan kunci dengan panjang 16 karakter. Plainteks dan kunci diintegrasi menjadi plainteks operasi proses.

Proses penyelesaian metoda kriptografi IDEA ini dapat dibagi menjadi 3 tahapan yaitu :

1. Proses Pembentukan Kunci.
2. Proses Enkripsi.
3. Proses Dekripsi.

Proses Pembentukan Kunci

Metoda IDEA memiliki *input* 128 bit kunci (*key*) yang identik dengan 32 digit heksadesimal ataupun 16 karakter yang diproses untuk menghasilkan 52 buah *subkey* dengan perincian masing-masing 6 buah *subkey* akan digunakan pada 8 putaran dan 4 buah *subkey* untuk transformasi *output*. Sebanyak 52 sub-blok kunci 16-bit untuk proses enkripsi diperoleh dari sebuah kunci 128-bit pilihan pemakai. Blok kunci 128-bit dipartisi menjadi 8 sub-blok kunci 16-bit yang lansung dipakai sebagai 8 sub-blok kunci pertama. Kemudian blok kunci 128-bit dirotasi dari kiri 25 posisi untuk dipartisi lagi menjadi 8 sub-blok kunci 16-bit berikutnya. Proses rotasi dan pertisi itu diulangi lagi sampai diperoleh 52 sub-blok kunci 16-bit, dengan urutan seperti pada tabel.

Tabel 1. Sub-blok Kunci Enkripsi

Proses	Sub-blok Kunci
Putaran Ke-1	Z11 Z21 Z31 Z41 Z51 Z61
Putaran Ke-2	Z12 Z22 Z32 Z42 Z52 Z62
Putaran Ke-3	Z13 Z23 Z33 Z43 Z53 Z63
Putaran Ke-4	Z14 Z24 Z34 Z44 Z54 Z64
Putaran Ke-5	Z15 Z25 Z35 Z45 Z55 Z65
Putaran Ke-6	Z16 Z26 Z36 Z46 Z56 Z66
Putaran Ke-7	Z17 Z27 Z37 Z47 Z57 Z67

Putaran Ke-8	$Z_{18}\ Z_{28}\ Z_{38}\ Z_{48}\ Z_{58}\ Z_{68}$
Transformasi *output*	$Z_{19}\ Z_{29}\ Z_{39}\ Z_{49}$

Keterangan :

1. Putaran ke-1 sampai dengan putaran ke-8 terdiri dari 6 buah sub-blok kunci dan putaran terakhir merupakan transformasi *output* yang terdiri dari 4 buah sub-blok kunci. Di mana kunci yang dimaksudkan di atas adalah (Z).
2. Putaran ke-1, Sub-blok kunci Z_{11} menyatakan sub-blok kunci pertama untuk putaran ke-1, Z_{21} menyatakan sub-blok kunci kedua untuk putaran pertama, Z_{31} menyatakan sub-blok kunci ketiga untuk putaran pertama, Z_{41} menyatakan sub-blok kunci keempat untuk putaran pertama, Z_{51} menyatakan sub-blok kunci kelima untuk putaran pertama, Z_{61} menyatakan sub-blok kunci keenam untuk putaran pertama demikian seterusnya sampai putaran ke-8 dan untuk transformasi *output.*

Tabel 2. Sub-blok Kunci Dekripsi

Proses	Sub-blok Kunci
Putaran Ke-1	$(Z_{19}^{-1})\ -Z_{29}\ -Z_{39}\ (Z_{49}^{-1})\ Z_{58}\ Z_{68}$
Putaran Ke-2	$(Z_{18}^{-1})\ -Z_{38}\ -Z_{28}\ (Z_{48}^{-1})\ Z_{57}\ Z_{67}$
Putaran Ke-3	$(Z_{17}^{-1})\ -Z_{37}\ -Z_{27}\ (Z_{47}^{-1})\ Z_{56}\ Z_{66}$
Putaran Ke-4	$(Z_{16}^{-1})\ -Z_{36}\ -Z_{26}\ (Z_{46}^{-1})\ Z_{55}\ Z_{65}$
Putaran Ke-5	$(Z_{15}^{-1})\ -Z_{35}\ -Z_{25}\ (Z_{45}^{-1})\ Z_{54}\ Z_{64}$
Putaran Ke-6	$(Z_{14}^{-1})\ -Z_{34}\ -Z_{24}\ (Z_{44}^{-1})\ Z_{53}\ Z_{63}$
Putaran Ke-7	$(Z_{13}^{-1})\ -Z_{33}\ -Z_{23}\ (Z_{43}^{-1})\ Z_{52}\ Z_{62}$
Putaran Ke-8	$(Z_{12}^{-1})\ -Z_{32}\ -Z_{22}\ (Z_{42}^{-1})\ Z_{51}\ Z_{61}$
Transformasi *output*	$(Z_{11}^{-1})\ -Z_{21}\ -Z_{31}\ (Z_{41}^{-1})$

Keterangan :

1. Untuk Putaran ke-1 sampai dengan putaran ke-8 terdiri dari 6 buah sub-blok kunci dan putaran terakhir merupakan transformasi *output* yang terdiri dari 4 buah sub-blok kunci.
2. Putaran ke-1 adalah 6 buah sub-blok kunci di mana 4 buah sub-blok kunci transformasi *output* dan 2 buah sub-blok kunci paling akhir putaran ke-8 dari sub-blok kunci enkripsi.
3. Untuk putaran ke-2 adalah 6 buah sub-blok kunci di mana 4 buah sub-blok kunci hasil putaran 8 dan 2 buah sub-blok kunci paling akhit putaran ke-7 dari sub-blok kunci dekripsi. Demikian seterusnya sampai putaran ke-8. Sedangkan 4 buah sub-blok kunci terakhir merupakan transformasi *output* untuk dekripsi.
4. Untuk putaran ke-1 sub-blok kunci pertama, keempat diinverskan $(Z_{19})^{-1}$, $(Z_{49})^{-1}$ dan sub-blok kunci kedua, ketiga menggunakan operator min $-Z_{19}$, $-Z_{49}$. Sedangkan 2 sub-blok kunci terakhir diambil dari sub-blok kunci enkripsi pada tabel 4.1. Demikian seterusnya sampai putaran ke-8.

Untuk lebih memahami proses pembentukan kunci pada metoda IDEA, diberikan sebuah simulasi berikut ini. Misalkan : *Input* kunci = '**KRIPTOGRAFI IDEA**'. Kunci yang diinputkan dikonversikan ke dalam biner. Proses pembentukan kuncinya adalah sebagai berikut :

KUNCI ENKRIPSI

PUTARAN – 1, kunci dirubah ke dalam bilangan biner.

INPUT KUNCI 128 bit

01001011010100100100100101010000010101000100111101000111010100100100000010100011
00100100100100000010010010100010001000101010000001

Kunci Pecah menjadi 8 kelompok :

Z_{11} (Putaran 1) = 0100101101010010
Z_{21} (Putaran 1) = 0100100101010000
Z_{31} (Putaran 1) = 0101010001001111
Z_{41} (Putaran 1) = 0100011101010010
Z_{51} (Putaran 1) = 0100000101000110
Z_{61} (Putaran 1) = 0100100100100000
Z_{21} (Putaran 2) = 0100100101000100
Z_{22} (Putaran 2) = 0100010101000001

Kemudian Left Shift, kunci 128-bit dirotasi dari kiri 25. Hasil proses di atas dirotasi dari kiri 25 posisi untuk dipartisi lagi menjadi 8 sub-blok kunci 16-bit berikutnya. Sedangkan untuk 4 sub kunci terakhir marupakan transformasi *output* yaitu :

Z_{19} (Transformasi *Output*) = 0101010000010101
Z_{29} (Transformasi *Output*) = 0001001111010001
Z_{39} (Transformasi *Output*) = 1101010010010000
Z_{49} (Transformasi *Output*) = 0101000110010010

KUNCI DEKRIPSI

Untuk proses sub-blok kunci dekripsi dilakukan proses berikut :

PUTARAN – 1

KD1(Putaran 1) = Inverse(KE1-Putaran9) = 1000000100010001
KD2(Putaran 1) = Minus(KE2-Putaran9) = 1110110000101111
KD3(Putaran 1) = Minus(KE3-Putaran9) = 0010101101110000
KD4(Putaran 1) = Inverse(KE4-Putaran9) = 1110100111000111
KD5(Putaran 1) = (KE5-Putaran8) = 0000100100101000
KD6(Putaran 1) = (KE6-Putaran8) = 1000100010101000

Untuk putaran berikutnya proses pembentukan kunci akan melakukan tahapan-tahapan seperti pada putaran pertama. Dan untuk 4 sub kunci terakhir marupakan transformasi *output* yaitu :

KD1(Putaran 9) = Inverse(KE1-Putaran1) = 0111010101101111
KD2(Putaran 9) = Minus(KE2-Putaran1) = 1011011010110000
KD3(Putaran 9) = Minus(KE3-Putaran1) = 1010101110110001
KD4(Putaran 9) = Inverse(KE4-Putaran1) = 1001111100011010

3.2.2 Proses Input Plainteks Dalam Enkripsi

Proses enkripsi dari metoda IDEA terdiri dari 8 iterasi (putaran) ditambah satu putaran transformasi *output*. Proses ini memiliki *input* data *plaintext* 64 bit yang identik dengan 16 digit heksadesimal atau 8 karakter.

Blok pesan terbuka dengan lebar 64-bit , X, dibagi menjadi 4 sub-blok 16-bit, X_1, X_2, X_3, X_4, sehingga X = (X_1, X_2, X_3, X_4). Keempat sub-blok 16-bit itu ditransformasikan menjadi sub-blok 16-bit, Y_2, Y_2, Y_3, $Y_{4,}$ sebagai pesan rahasia 64-bit Y = (Y_1, Y_2, Y_3, Y_4) yang berada di bawah kendali 52 sub-blok kunci 16-bit yang dibentuk dari dari blok kunci 128 bit.

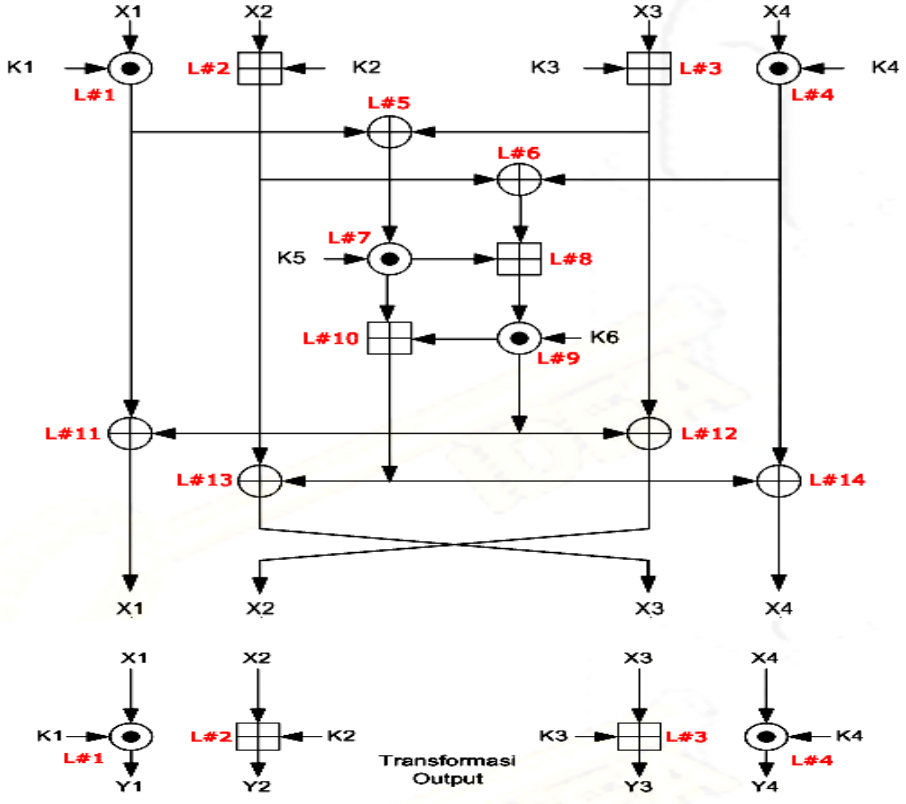
Keempat sub-blok 16-bit, X_1, X_2, X_3, X_4, digunakan sebagai masukan untuk putaran pertama dari algoritma IDEA. Dalam setiap putaran dilakukan operasi XOR, penjumlahan,

perkalian antara dua sub-blok 16-bit dan diikuti pertukaran antara sub-blok 16-bit putaran kedua dan ketiga. Keluaran putaran sebelumnya menjadi masukan putaran berikutnya. Setelah putaran kedelapan dilakukan transformasi keluaran yang dikendalikan oleh 4 sub-blok kunci 16-bit.

Pada setiap putaran dilakukan operasi-operasi sebagai berikut :

1. Perkalian X_1 dengan sub-kunci pertama mod (2^16 + 1).
2. Penjumlahan X_2 dengan sub-kunci kedua mod (2^16).
3. Pejumlahan X_3 dengan sub-kunci ketiga mod (2^16).
4. Perkalian X_4 dengan sub-kunci keempat mod (2^16 + 1).
5. Operasi XOR hasil langkah 1 dan 3
6. Operasi XOR hasil angkah 2 dan 4
7. Perkalian hasil langkah 5 dengan sub-kunci kelima mod (2^16 + 1).
8. Penjumlahan hasil langkah 6 dengan langkah 7 mod (2^16).
9. Perkalian hasil langkah 8 dengan sub-kunci keenam mod (2^16 + 1).
10. Penjumlahan hasil langah 7 dengan 9
11. Operasi XOR hasil langkah 1 dan 9
12. Operasi XOR hasil langkah 3 dan 9
13. Operasi XOR hasil langkah 2 dan 10
14. Operasi XOR hasil langkah 4 dan 10

Di mana diagram bloknya sebagai berikut :



Gambar 4. Diagram Blok Enkripsi

Keluaran setiap putaran adalah 4 sub-blok yang dihasilkan pada langkah 11, 12, 13, dan 14 dan menjadi masukan putaran berikutnya.

Setelah putaran kedelapan terdapat transformasi keluaran, yaitu :

1. Perkalian X_1 dengan sub-kunci pertama
2. Penjumlahan X_2 dengan sub-kuci ketiga

3. Penjumlahan X_3 dengan sub-kunci kedua
4. Perkalian X_4 dengan sub-kunci keempat

Terakhir, ke-empat sub-blok 16-bit yang merupakan hasil operasi 1, 2, 3, dan 4 digabung kembali menjadi blok pesan rahasia (chiperteks) 64-bit.

Untuk proses enkripsi IDEA dapat dilihat pada contoh berikut ini :

Hasil enkripsi tiap putaran yang diproses dengan algoritma IDEA untuk sebuah pesan terbuka (plainteks)=**'HARVEIDH'** yang telah dibagi menjadi empat bagian yaitu X_1 =**HA**, X_2 =**RV**, X_3 =**EI**, dan X_4 =**DH**, dan kunci=**'KRIPTORAFI IDEA'** di mana kunci telah dibagi-bagi menjadi Z_{11} =**KR**, Z_{21} =**IP**, Z_{31} =**TO**, Z_{41} =**GR**, Z_{51} =**AF**, Z_{61} =**I** , Z_{12} =**ID**, Z_{22} =**EA**.

Konversikan Plainteks Ke biner dan hexadecimal berdasarkan tabel ASCII (Lampiran D). Konveri karakter dapat dilihat pada tabel.

Tabel 4. Konversi Karakter Ke Biner Dan Hexadecimal

Karakter	Biner	Hexadecimal
H	01001000	048
A	01000001	041
R	01010010	052
V	01010110	056
E	01000101	045
I	01001001	049
D	01000100	044
H	01001000	048

Maka proses enkripsinya dalam setiap putaran sebagai berikut :

PUTARAN – 1 DALAM BENTUK BINER, terdapat 14 langkah sebagai berikut:

Langkah-1 = (X1 * K1) mod (2^16 + 1) = 0001101010010000
Langkah-2 = (X2 + K2) mod 2^16 = 1001101110100110
Langkah-3 = (X3 + K3) mod 2^16 = 1001100110011000
Langkah-4 = (X4 * K4) mod (2^16 + 1) = 1100010000001011
Langkah-5 = langkah-1 XOR Langkah-3 = 1000001100001000
Langkah-6 = langkah-2 XOR Langkah-4 = 0101111110101101
Langkah-7 =(langkah-5 * K5) mod (2^16 + 1) = 1011101011001000
Langkah-8 =(langkah-6 + langkah-7) mod 2^16 = 0001101001110101
Langkah-9 =(langkah-8 * K6) mod (2^16 + 1) = 1010010000010010
Langkah-10 = (langkah-7 + langkah-9)) mod 2^16=0101111011011010
Langkah-11 = langkah-1 XOR langkah-9 = 1011111010000010
Langkah-12 = langkah-3 XOR langkah-9 = 0011110110001010
Langkah-13 = langkah-2 XOR langkah-10 = 1100010101111100
Langkah-14 = langkah-4 XOR langkah-10 = 1001101011010001

Hasil putaran pertama adalah sebagai berikut :

X1 = Langkah 11 = 1011111010000010
X2 = Langkah 12 = 0011110110001010
X3 = Langkah 13 = 1100010101111100
X4 = Langkah 14 = 1001101011010001

Hasil putaran pertama yaitu langkah 11, langkah 12, langkah 13 dan langkah 14 merupakan masukan untuk putaran berikutnya, demikian seterusnya. Untuk putaran berikutnya melakukan tahap-tahap proses seperti langkah pertama, sehingga didapat hasilnya sebagai berikut :

PUTARAN – 2,

X1 = Langkah 11 = 1010100011100010
X2 = Langkah 12 = 1011100011000101

X3 = Langkah 13 = 1000011010000101
X4 = Langkah 14 = 0111101001000010

PUTARAN - 3

X1 = Langkah 11 = 1011101011010000
X2 = Langkah 12 = 0001010110000111
X3 = Langkah 13 = 1010001000010010
X4 = Langkah 14 = 0101000001010000

Setelah putaran 4 sampai 8, maka dilakukan transpormasi output.

TRANSFORMASI *OUTPUT*

01) Y1 = (X1 * K1) mod (2^16 + 1) = 0000111110001111
02) Y2 = (X2 + K2) mod 2^16 = 1001101100111011
03) Y3 = (X3 + K3) mod 2^16 = 1001010111110100
04) Y4 = (X4 * K4) mod (2^16 + 1) = 0110010111110111

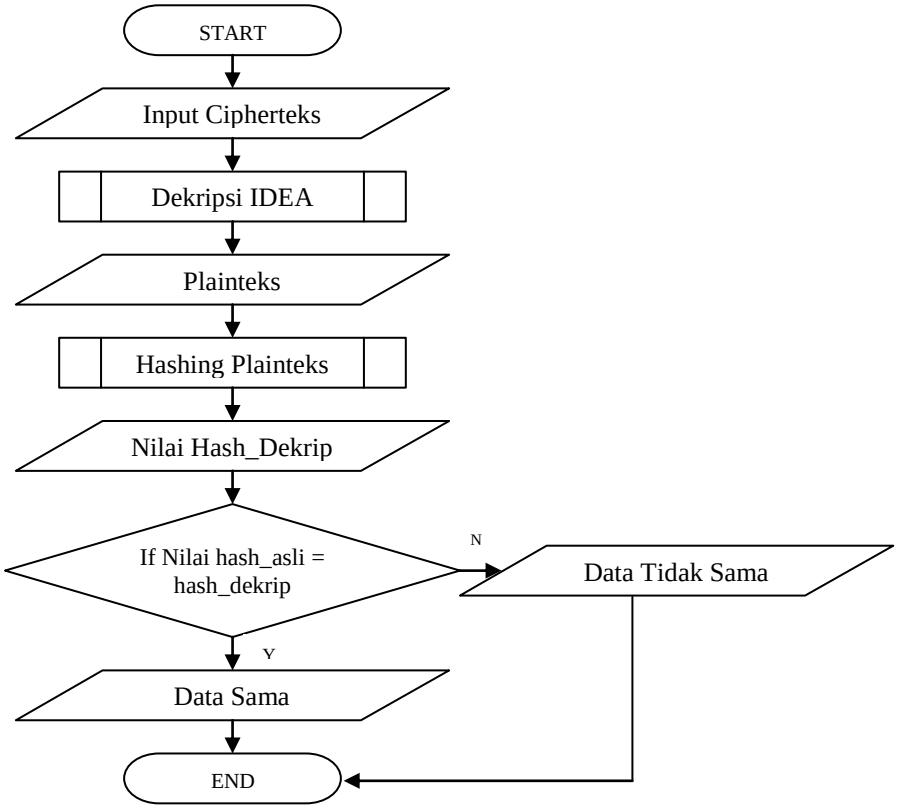
Setelah melakukan proses 8 putaran dan transformasi *output* di atas maka didapat hasil enkripsinya sebagai berikut :

Hasil Enkripsi :

Y1 = 0000111110001111 = • •
Y2 = 1001101100111011 = ›;
Y3 = 1001010111110100 = •ô
Y4 = 0110010111110111 = e÷
Ciphertext = □ □ ›;•ôe÷

Analisa Proses Dekripsi

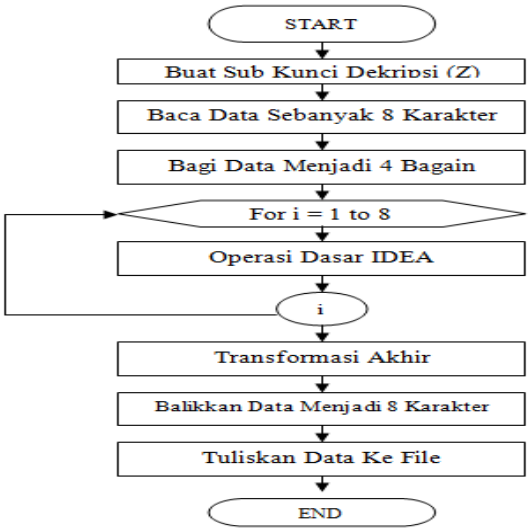
Proses dekripsi merupakan sistem untuk mengolah data acak (cipherteks) menjadi data awal (plainteks). Dalam proses dekripsi ini terdapat proses dekripsi IDEA dan proses cek *hashing* plainteks. Secara umum proses kerja dekripsi dapat digambarkan seperti gambar.



Gambar 5. *Flowchart* Proses Dekripsi

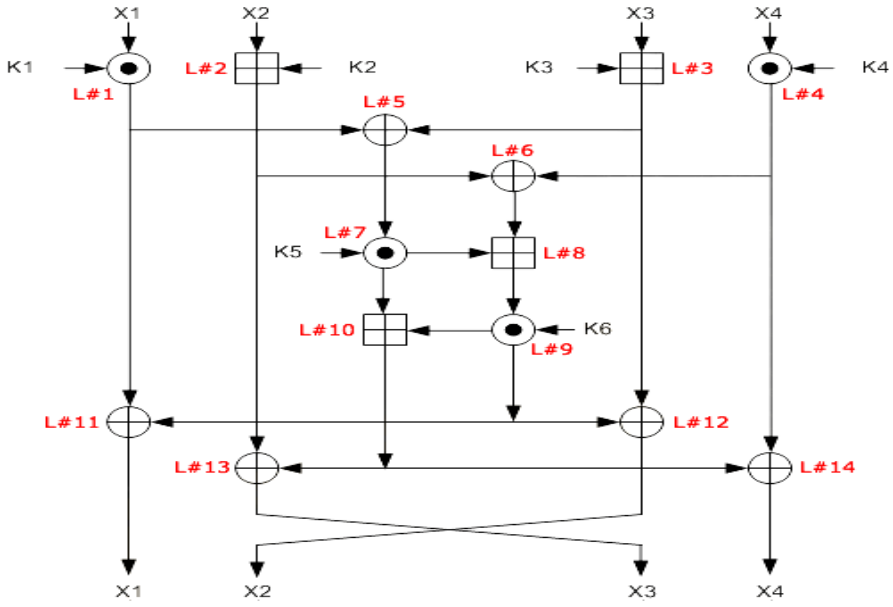
Dekripsi IDEA

Proses dekripsi dilakukan dengan cara yang sama dengan proses enkripsi kecuali pembentukan dari *subkey*. Inputan yang dibutuhkan adalah chiperteks dan kunci dekripsi yang didapat dari sub kunci enkripsi Z. Pembacaan pada dokumen dilakukan tiap 64 bit (X) seperti pada proses enkripsi dan bersama dengan sub kunci dekripsi (U) untuk masukan iterasi pertama. Iterasi dilakukan sebanyak delapan kali. Proses di atas dapat ditunjukkan pada gambar.



Gambar 6. *Flowchart* Dekripsi IDEA

Proses dekripsi dapat dilihat pada diagram blok gambar.



Gambar 7. Diagram Blok Dekripsi IDEA

Dengan menggunakan kunci yang diturunkan dari kunci enkripsi dan dengan menggunakan blok dekripsi yang sama dengan proses enkripsi. Hasil dekripsi tiap putaran yang diproses dengan algoritma IDEA untuk pesan yang dienkripsi (chiperteks)=**”□ □ ›;•ôe”** yang telah dibagi menjadi empat bagian yaitu X_1 =**• •**, X_2 =**›;**, X_3 =**•ô**, dan X_4 =**e÷**, denga kunci yang sama yaitu **‘KRIPTORAFI IDEA’**. Maka diperoleh proses putaran dekripsinya sebagai berikut :

<u>PUTARAN – 1 DALAM BENTUK BINER</u>, terdapat 14 langkah sebagai berikut:

01) Langkah 1 = (X1 * K1) mod (2^16 + 1) = 0000111110100111

02) Langkah 2 = (X2 + K2) mod 2^16 = 1000011101101010
03) Langkah 3 = (X3 + K3) mod 2^16 = 1100000101100100
04) Langkah 4 = (X4 * K4) mod (2^16 + 1) = 1011010011100101
05) Langkah 5 = Langkah 1 XOR Langkah 3 = 1100111011000011
06) Langkah 6 = Langkah 2 XOR Langkah 4 = 0011001110001111
07) Langkah 7 = (Langkah 5 * K5) mod (2^16 + 1) = 0010001000010011
08) Langkah 8 = (Langkah 6 + Langkah 7) mod 2^16 = 0101010110100010
09) Langkah 9 = (Langkah 8 * K6) mod (2^16 + 1) = 0001010010011010
10) Langkah 10 =(Langkah 7 + Langkah 9) mod 2^16 = 0011011010101101
11) Langkah 11 = Langkah 1 XOR Langkah 9 = 0001101100111101
12) Langkah 12 = Langkah 3 XOR Langkah 9 = 1101010111111110
13) Langkah 13 = Langkah 2 XOR Langkah 10 = 1011000111000111
14) Langkah 14 = Langkah 4 XOR Langkah 10 = 1000001001001000

Hasil putaran pertama adalah sebagai berikut :

X1 = Langkah 11 = 0001101100111101
X2 = Langkah 12 = 1101010111111110
X3 = Langkah 13 = 1011000111000111
X4 = Langkah 14 = 1000001001001000

Hasil putaran pertama yaitu langkah 11, langkah 12, langkah 13 dan langkah 14 merupakan masukan untuk putaran berikutnya, demikian seterusnya. Untuk putaran berikutnya melakukan tahap-tahap proses yang sama seperti langkah pertama, sehingga didapat hasilnya sebagai berikut :

PUTARAN - 2

X1 = L#11 = 1011101111110010
X2 = L#12 = 1010100111100001
X3 = L#13 = 0110100011111000
X4 = L#14 = 1100001110010000

Setelah putaran 3 sampai 8 akan didapatkan transformasi output.

TRANSFORMASI *OUTPUT*

01) Y1 = (X1 * K1) mod (2^16 + 1) = 0100100001000001
02) Y2 = (X2 + K2) mod 2^16 = 0101001001010110
03) Y3 = (X3 + K3) mod 2^16 = 0100010101001001
04) Y4 = (X4 * K4) mod (2^16 + 1) = 0100010001001000

Setelah melakukan proses 8 putaran dan transformasi *output* di atas maka didapat hasil dekripsinya sebagai berikut :

Hasil Dekripsi :

Y1 = 0100100001000001 = HA
Y2 = 0101001001010110 = RV
Y3 = 0100010101001001 = EI
Y4 = 0100010001001000 = DH
Plaintext = HARVEIDH

SIMPULAN

Setelah melalui akhir proses penelitian ini, penulis menarik kesimpulan sebagai berikut :

1. Fungsi *hash* adalah fungsi yang bisa digunakan untuk berbagai keperluan. Fungsi *hash* dalam kriptografi memiliki beberapa sifat tambahan yang dapat digunakan dalam pengamanan data. Jika dalam database fungsi *hash* digunakan untuk memudahkan penyimpanan key dalam tabel *hash*, pada kriptografi, fungsi *hash* digunakan untuk memastikan kebenaran pesan yang dikirim dengan cara membandingkan nilai-nilai *hash* yang diperoleh.

2. Penggunaan fungsi *hash* pada algoritma enkripsi IDEA secara bersamaan akan meningkat-kan *Confidentiality* dan *Integrity* data.
3. Dari hasil pengujian kecepatan proses dapat disimpulkan bahwa sistem dapat diaplikasikan pada dunia nyata.

DAFTAR RUJUKAN

[1] **Ariyus, Dony**. "*Kriptografi Keamanan Data Dan Komunikasi*". Edisi Pertama. Yogyakarta. Graha Ilmu. 2005.

[2] **Stallings,William**. "*Cryptography and Network Security Principles and Practices*". Fourth Edition. Prentice Hall. 2005.

[3] **Egie Wendra Apriliawan**. "*Fungsi Hash Pada Kriptografi*". Program Studi Ilmu Komputer Universitas Pendidikan Indonesia Bandung Jl. Setiabudhi No.229 Bandung Indonesia. E-mail: egiewendra@gmail.com

[4] **Radius Indrawan, Justinus Andjarwirawan, Gregorius S. Budhi. "***Fungsi Hash SNEFRU dan Metode Enkripsi IDEA Untuk Keamanan Dokumen Elektronik"*. Dosen Fakultas Teknologi Industri. Jurusan Teknik Informatika. Universitas Kristen Petra. Email: greg@petra.ac.id.

[5] **Rezza Mahyudin – NIM : 13505055**. *"Algoritma Message Digest 5 (MD5) Dalam Aplikasi Kriptografi"*. Program Studi Teknik Informatika. Institut Teknologi Bandung Jl. Ganesha 10, Bandung E-mail : if15055@students.if.itb.ac.id.

[6] **Tiffany Adriana – NIM 13505068**. "*Kriptografi Dan Pemanfaatannya Pada Rsa Dan Md5*". Program Studi Teknik Informatika Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung Jl. Ganesha 10, Bandung. E-mail : if15068@students.if.itb.ac.id

[7] **Sibghatullah Mujaddid (13507124). "***Kriptoanalisis Pada Fungsi Hash Kriptografi MD5***".** Jurusan Teknik Informatika ITB, Bandung 40132. E-mail: sibgha07@students.itb.ac.id.

[8] **Brian Al Bahr – NIM: 13506093**. "*Lux Hash Function***".** Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung Jl. Ganesha 10, Bandung. E-mail: if16093@students.if.itb.ac.id, bhbrayeun@yahoo.co.id.

[9] **Andrew Regenscheid, Ray Perlner, Shu-jen Chang, John Kelsey, Mridul Nandi, Souradyuti Paul**. "*Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competition"*. Information Technology Laboratory National Institute of Standards and Technology Gaithersburg, MD 20899-8930.

[10] **Stevens Jethefer – 13504080**. "*Studi Dan Perbandingan Algoritma Idea (International Data Encryption Algorithm) Dengan Des (Data Encryption Standard)"*. Program Studi Teknik Informatika. Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung Jl. Ganeca 10. Bandung. E-mail : if14080@students.if.itb.ac.id.