ISSN : 1978-6603

JURNAL Ilmiah SAINTIKOM Sains dan Komputer

PERANGKAT LUNAK PEMBELAJARAN KRIPTOGRAFI METODE WAKE (*WORD AUTO KEY ENCRYPTION*)

Muhammad Dahria[#1], Abdul Rahim[#2], Hendra Jaya[#3]

[#1,2,3] Program Studi Sistem Informasi, STMIK Triguna Dharma
Jl. A.H. Nasution No. 73 F - Medan
E-mail : [#1]m.dahria@gmail.com

Abstrak

Kriptografi digunakan untuk mengamankan data yang bersifat rahasia agar data tersebut tidak diketahui oleh orang lain yang tidak berkepentingan. Dalam ilmu kriptografi, masih banyak metode yang dapat digunakan untuk mengamankan data. Setiap metode memiliki kelebihan dan kekurangannya masing-masing. Namun, yang menjadi permasalahan dalam memilih metode kriptografi yang cocok adalah bagaimana mengetahui dan memahami cara kerja dari metode kriptografi tersebut. Metode WAKE (Word Auto Key Encryption) Proses penyelesaian metoda ini cukup rumit dan sulit untuk dikerjakan secara manual karena algoritmanya yang cukup panjang dan kompleks. Metode WAKE, dapat dibagi menjadi beberapa proses yaitu proses pembentukan tabel dan kunci, enkripsi dan dekripsi. Untuk memudahkan pemahaman cara kerja Metode WAKE, diperlukan sebuah perangkat lunak. Dan metode ini cukup cepat dalam implementasinya pada perangkat lunak, dan dapat menjelaskan langkah-langkah maupun hasil setiap langkah, untuk pembelajaran metode kriptografi WAKE.

Kata Kunci : kriptografi, perangkat lunak, metode WAKE.

Abstract

Cryptography is used to secure confidential data so that the data is not known by other people who are not interested. In the science of cryptography, there are many methods that can be used to secure the data. Each method has advantages and disadvantages of each. However, the problem of choosing a suitable method of cryptography is to know and understand the workings of the cryptographic methods. Methods WAKE (Word Auto Key Encryption) method of settlement process is quite complicated and difficult to do manually because the algorithm is quite long and complex. WAKE methods, can be divided into several processes, namely the formation of tables and keys, encryption and decryption. To facilitate the understanding of the workings of the method WAKE, needed a software. And this method is quite fast in its implementation in the software, and can explain the steps and results of each step, learning WAKE cryptographic methods.

***Keywords** : cryptography, software, methods WAKE.*

PENDAHULUAN

Saat ini media digital seperti teks, video, audio, dan gambar telah menggantikan peran media analog dalam berbagai aplikasi. Hal ini disebabkan karena beberapa kelebihan yang dimiliki media digital seperti transmisi yang bebas derau, penyimpanan yang padat, penyalinan yang sempurna, dan kemudahan untuk melakukan pengeditan. Disamping kelebihan yang dimiliki oleh media digital, terdapat kelemahan dari penggunaan media digital. Masalah terbesar adalah mengenai hak intelektual (hak cipta) dan kebenaran konten dari suatu media digital. Diperlukan suatu mekanisme yang dapat mengamankan data agar aspek kerahasiaan (*confidentiality)* agar dapat tetap terjaga.

Metode kriptografi dapat digunakan untuk mengamankan data yang bersifat rahasia agar data tersebut tidak diketahui oleh orang lain yang tidak berkepentingan. Metode WAKE merupakan salah satu metode yang telah digunakan secara komersial. WAKE merupakan singkatan dari *Word Auto Key Encryption*. Metode ini ditemukan oleh David Wheeler pada tahun 1993. Metode ini menggunakan kunci 128 bit, dan sebuah tabel 256 x 32 bit. Dalam algoritmanya, metode ini menggunakan operasi XOR, AND, OR dan *Shift Right*. Metode WAKE ini telah digunakan pada *program* Dr. Solomon Anti Virus versi terbaru. Metode WAKE dapat dibagi menjadi beberapa proses yaitu proses pembentukan tabel dan kunci, enkripsi dan dekripsi. Proses penyelesaian metode ini cukup rumit dan sulit untuk dikerjakan secara manual berhubung karena algoritmanya yang cukup panjang dan kompleks.

Dalam ilmu kriptografi, selain metode WAKE, masih banyak metode yang dapat digunakan untuk mengamankan data. Setiap metode memiliki kelebihan dan kekurangannya masing-masing. Namun, yang menjadi permasalahan dalam memilih metode kriptografi yang cocok adalah bagaimana mengetahui dan memahami cara kerja dari metode kriptografi tersebut. Oleh karena itu, diperlukan suatu perangkat lunak untuk mempelajari metode kriptografi tersebut. Penulis memilih metode WAKE karena metode ini cukup cepat dalam implementasinya pada perangkat lunak.

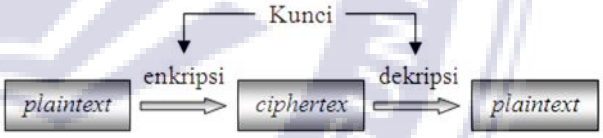
LANDASN TEORI

1. Sistem Kriptografi

Berdasarkan jumlah kunci yang digunakan, ada dua jenis sistem kriptografi yaitu sistem kriptografi simetris dan sistem kriptografi asimetris.

a. Sistem Kriptografi Simetris

Enkripsi simetris sering juga disebut sebagai enkripsi konvensional atau enkripsi kunci-tunggal (single key). Pada model enkripsi simetris ini digunakan algoritma yang sama untuk proses enkripsi/dekripsi dengan memakai satu kunci yang sama.



Gambar 1. Model Sistem Kriptografi Simetris

Keamanan dari enkripsi simetris bergantung pada beberapa faktor, yaitu :

1. Algoritma enkripsi harus cukup kuat sehingga tidaklah praktis untuk mendekripsi suatu pesan hanya dengan memiliki *cyphertext* saja.
2. Keamanan dari enkripsi simetris adalah bergantung pada kerahasiaan kunci, bukan kerahasiaan dari algoritma enkripsi itu sendiri. Semakin panjang kunci yang dipakai maka semakin sulit untuk menebak kunci dengan menggunakan metode *brute*

force attacks (mencoba semua kemungkinan kunci).

Algoritma enkripsi simetris yang popular dewasa ini adalah DES (*Data Encryption Standard*) dengan panjang kunci 56-bit, IDEA (128-bit), Twofish (sampai dengan 256-bit), Rijndael (sampai dengan 256-bit) dan lain-lain.

b. Sistem Kriptografi Asimetris

Sistem kriptografi asimetris biasanya lebih dikenal dengan kriptografi kunci-publik (*public-key cryptography)*. Ide kriptografi asimetris ini pertama kali dimunculkan oleh Whitfield Diffie dan Martin Hellman pada tahun 1976. Diffie dan Hellman mempostulatkan sistem ini tanpa menunjukkan algoritmanya. Walaupun demikian mereka menjabarkan syarat-syarat yang harus dipenuhi oleh suatu algoritma *public-key* yaitu :

1. Mudah secara komputasi bagi suatu pihak B untuk mengkonstruksi sepasang kunci asimetris (kunci public KU, kunci pribadi KR).
2. Mudah secara komputasi bagi pengirim A, dengan memiliki kunci public B dan pesan yang ingin dienkripsi, M, untuk menghasilkan *ciphertext* (C) :

$$C = E_{KUb}(M)$$

3. Mudah secara komputasi bagi penerima B untuk mendekripsi *ciphertext* yang dihasilkan dengan menggunakan kunci pribadinya untuk mengembalikan pesan aslinya.

$$M = D_{KRb}(C) = D_{KRb}[E_{KUb}(M)]$$

4. Tidak bisa secara komputasi bagi pihak ketiga untuk memperoleh kunci pribadi KRb hanya dengan mengetahui kunci public KUb.
5. Tidak bisa secara komputasi bagi pihak ketiga untuk mengembalikan data asli M hanya dengan mengetahui kunci public KUb dan *ciphertext* C.

   Walaupun bukanlah suatu keharusan bagi semua aplikasi *public-key*, namun persyaratan keenam bisa ditambahkan :
6. Fungsi enkripsi dan dekripsi bisa diterapkan dengan urutan yang dibalik :

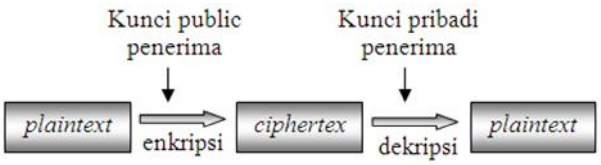
$$M = E_{KUb}[D_{KRb}(M)]$$

Kegunaan dari persyaratan keenam adalah untuk penerapan tanda tangan digital (*digital signature*) yang digunakan memecahkan isu otentikasi (*authentication*) dalam masalah keamanan data.

Menurut Stalling, proses enkripsi *public-key* sederhana melibatkan empat tahap berikut :

1. Setiap *user* di dalam jaringan membuat sepasang kunci untuk digunakan sebagai kunci enkripsi dan dekripsi dari pesan yang akan diterima.
2. *User* mempublikasikan kunci enkripsinya dengan menempatkan kunci publiknya ke tempat umum. Pasangan kunci yang lain tetap dijaga kerahasiaannya.
3. Jika *user* A ingin mengirimkan sebuah pesan ke *user* B, ia akan mengenkripsi pesan tersebut dengan menggunakan kunci publik *user* B.

Pada saat *user* B ingin mengirimkan sebuah pesan ke *user* B, ia akan menggunakan kunci pribadinya sendiri. Tidak ada pihak lain yang bisa mendekripsi pesan itu karena hanya B sendiri yang mengetahui kunci pribadi B.



Gambar 2. Model Sistem Kriptografi Asimetris

Sampai saat ini hanya ada beberapa sistem *cryptography* asimetris yang dipublikasikan. Yang paling berhasil sejauh ini adalah algoritma RSA yang memenuhi keenam persyaratan *public-key* di atas. Di samping itu, algoritma enkripsi *public-key* yang lain adalah LUC, DSS, Diffie-Hellman dan lain-lain.

Kunci publik adalah kunci yang tidak disembunyikan dan boleh diketahui oleh orang lain. Kunci publik digunakan dalam proses enkripsi.

Kunci *private* adalah kunci rahasia yang tidak boleh diketahui oleh orang lain. Kunci *private* digunakan dalam proses dekripsi.

2. Aplikasi Kriptografi

a. Privacy

Privacy (kerahasiaan) mungkin merupakan aplikasi paling nyata dari kriptografi. Kriptografi dapat digunakan untuk mengimplementasikan *privacy* hanya dengan mengenkrip informasi yang diinginkan untuk tetap *private*. Agar seseorang dapat membaca data *private* ini dia harus mendekrip terlebih dahulu. Kadang-kadang informasi tertentu bukan untuk diakses oleh siapapun juga, dan dalam hal ini informasi dapat disimpan sedemikian rupa sehingga membalik proses merupakan sesuatu yang secara virtual tidak mungkin. Misalnya, dalam sistem *multi-user*, tidak ada satu orangpun dimungkinkan untuk mengetahui daftar *password* dari masing-masing *user* dalam sistem. Biasanya nilai hash dari *password* yang disimpan bukan *password* itu sendiri. Hal ini memungkinkan *user* dari sistem yakin betul tentang informasi pribadi disimpan betul-betul aman dari gangguan orang lain karena dengan memasukkan *password* harus diverifikasi terlebih dahulu (dengan menghitung fungsi hashnya dan membandingkan dengan nilai hash yang tersimpan).

b. Digital Signature dan Authentication

Authentication adalah suatu proses untuk membuktikan dan memverifikasi informasi tertentu. Kadang-kadang seseorang ingin memverifikasi asal dokumen, indentitas pengirim, waktu dan tanggal penandatanganan dan/atau pengiriman, identitas komputer atau user dan lain-lain. Suatu *digital signature* adalah cara *cryptography* dimana dengan cara tersebut beberapa hal di atas dapat diverifikasi. Tanda tangan dijital dari suatu dokumen adalah potongan informasi yang didasarkan kepada dokumen dan kunci rahasia penanda-tangan. Tanda tangan ini biasanya diciptakan melalui penggunaan fungsi hash dan fungsi tanda tangan privat (enkripsi kunci rahasia penanda tangan), tetapi masih ada metode lain.

Setiap hari orang menandatangani nama mereka dalam surat, bukti penggunaan kartu kredit, dan dokumen lainnya, yang menunjukkan bahwa mereka setuju dengan isi dokumen tersebut. Dalam hal ini, mereka melakukan otentikasi bahwa mereka dalam kenyataannya adalah pemilik atau pengirim atau sumber dari dokumen. Hal ini memungkinkan orang lain untuk melakukan verifikasi bahwa pesan khusus benar-benar berasal dari si penanda-tangan dokumen. Akan tetapi, cara ini bukanlah bebas dari kemungkinan pencurian atau pemalsuan dari pihak ketiga, karena orang dapat "mengangkat" tanda-tangan dari dalam dokumen dan menempatkannya ke dokumen lain, dengan demikian menghasilkan dokumen asli tapi palsu (aspal). Tanda tangan konvensional (dengan tinta) juga tidak aman dari pemalsuan karena dimungkinkan untuk mereproduksi sebuah tanda tangan pada dokumen lain atau mengubah dokumen setelah dokumen ditanda-tangani. Tanda tangan dijital dan tulisan tangan tergantung pada fakta bahwa sulit untuk mendapatkan dua orang dengan tanda tangan sama.

c. Key Agreement Protocol

Suatu *key agreement protocol*, juga dikenal dengan *key exchange protocol*, adalah sebarisan langkah yang dilakukan bila dua atau lebih pihak perlu sepakat atas suatu kunci yang digunakan untuk suatu *secret-key cryptosystem*. Protokol ini memungkinkan orang menggunakan kunci secara bersama dengan bebas dan aman melalui suatu medium yang tidak aman, tanpa perlu terlebih dahulu ada pembentukan kunci rahasia bersama.

Misalkan Ali dan Bob ingin menggunakan satu *secret-key cryptosystem* untuk berkomunikasi secara aman. Mereka terlebih dahulu harus memutuskan apa kunci yang mereka gunakan. Bob bukannya menelepon Ali dan mendiskusikan apa kunci yang mereka gunakan, yang bisa saja didengar oleh pihak ketiga, mereka memutuskan menggunakan satu protokol kesepakatan kunci (*key agreement protocol*). Dengan menggunakan *key agreement protocol*, Ali dan Bob dapat saling mempertukarkan kunci dalam lingkungan yang tidak aman. Salah satu contoh protokol ini adalah Diffie-Hellman *key agreement*. Dalam beberapa kasus, *public-key cryptography* digunakan dalam suatu *key agreement protocol*. Contoh lainnya adalah penggunaan amplop dijital (*digital envelopes*) untuk *key agreement*.

d. Identification (Identifikasi)

Identification (identifikasi) adalah suatu proses melalui mana seseorang yakin tentang identitas orang lain atau entittas tertentu. Dalam kehidupan kita sehari-hari kita mengidentifikasikan anggota keluarga kita, kawan, dan teman sejawat dengan karakteristik fisik mereka, seperti suara, muka atau karakteristik lainnya. Karakteristik ini disebut *biometrics*, yang hanya dapat digunakan pada jaringan dengan perangkat khusus. Entitas dalam sebuah jaringan dapat juga mengidentifikasikan entitas lain dengan menggunakan metode kriptografi.

Otentikasi dan identifikasi adalah dua hal yang berbeda. Identifikasi mengharuskan *verifier* (pelaku verifikasi) membandingkan informasi yang diberikan terhadap semua entitas yang diketahuinya, sedangkan otentikasi membutuhkan pengecekan informasi tentang entitas tunggal yang diberikan dan diidentifikasikan sebelumnya. Selanjutnya, identifikasi harus mengidentifikasikan entitas secara unik, sementara autentikasi tidak mengharuskan keunikan. Sebagai contoh, seseorang yang sedang *log into* rekening bersama tidak diidentifikasikan secara unik, tetapi dengan mengetahui *password* bersama, mereka diotentikasikan sebagai salah satu pemilik/pengguna rekening tersebut. Kemudian, identifikasi tidak perlu mengotentikasikan pengguna dalam maksud tertentu.

3. Dasar Matematika Kriptografi

Beberapa operasi dasar matematika yang digunakan dalam kriptografi metode WAKE adalah operasi AND, OR, XOR, Penjumlahan Modulo dan Shift Right.

a. Operator AND

Operasi *AND* dari dua *input* A dan B hanya akan bernilai bit “1” apabila kedua bit *input* A dan B bernilai bit “1”. Atau dengan kata lain *output* dari operasi *AND* akan memiliki nilai bit “0” apabila salah satu *input*-nya bernilai bit “0”. Operasi AND dilambangkan dengan tanda “$\wedge$”.

Aturan operasi AND dapat dinyatakan seperti tabel berikut :

Tabel 1. Aturan operasi AND

A	B	A ^ B
0	0	0
0	1	0
1	0	0
1	1	1

Contoh :

11000110
10110011
---------- ∧
10000010

b. Operator OR

Operasi OR dari dua *input* A dan B hanya akan bernilai bit "0" apabila kedua bit *input* A dan B bernilai bit "0". Atau dengan kata lain *output* dari operasi OR akan memiliki nilai bit "1" apabila salah satu inputnya bernilai bit "1". Operasi OR dilambangkan dengan tanda "∨".

Aturan operasi *OR* dapat dinyatakan seperti tabel berikut :

Tabel 2. Aturan operasi AND

A	B	A ∨ B
0	0	0
0	1	1
1	0	1
1	1	1

Contoh :

11000110
10110011
---------- ∨
11110111

c. XOR

XOR adalah operasi *Exclusive*-OR yang dilambangkan dengan tanda "⊕". Hasil dari operasi XOR akan bernilai bit "0" (nol) jika dua buah bit *input* memiliki nilai yang sama dan akan menghasilkan nilai bit "1" (satu) jika dua buah bit *input* memiliki nilai bit yang berbeda. Aturan operasi XOR dapat dirumuskan seperti tabel berikut ini :

Tabel 3. Aturan operasi XOR

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Contoh :

11000110
10110011
---------- ⊕
01110101

d. Penjumlahan Modulo

Penjumlahan bit modulo yang digunakan dalam metode WAKE adalah penjumlahan dua buah bit bilangan yang sama panjang dan menghasilkan bilangan dengan panjang bit yang sama pula. Jika panjang bit bilangan lebih besar, maka bit bilangan yang berlebihan tersebut akan dibuang.

Contoh : 10111101 + 10010101 = 1 01010010

Bit 1 yang berlebihan di depan akan dibuang sehingga hasil yang didapatkan dari proses penjumlahan di atas adalah 01010010.

e. Pergeseran Bit (Shift)

Pergeseran bit (Shift) adalah operasi pergeseran terhadap suatu barisan bit

sebanyak yang diinginkan. Bit kosong yang telah tergeser akan diberikan nilai bit "0" (nol).

Operasi pergeseran terbagi menjadi dua macam yaitu :

1. Operasi Geser Kiri (*Shift Left*) yaitu operasi yang menggeser sejumlah bit ke kiri (*left*) dan mengisi tempat kosong dengan nilai bit "0" (nol). Operasi *shift left* dilambangkan dengan "**<<**". Contoh operasi *shift left* :

   11000110 << 1 : 10001100
   11000110 << 2 : 00011000
2. Operasi Geser Kanan (*Shift Right*) yaitu operasi yang menggeser sejumlah bit ke kanan (*right*) dan mengisi tempat kosong dengan nilai bit "0" (nol). Operasi *shift right* dilambangkan dengan "**>>**". Contoh operasi *shift right* : [3]

   11000110 >> 1 : 01100011
   11000110 >> 2 : 10110001

f. Konversi Bilangan Berbasis

Bilangan – bilangan berbasis dapat diubahatau dikonversikan satu sama lain. Proses pengubahan bilangan berbasis yang akan dibahas antara lain :

1. Pengubahan bilangan biner ke bilangan heksadesimal
2. Pengubahan bilangan heksadesimal ke bilangan biner

Konversi dari Bilangan Biner ke Bilangan Heksadesimal

Proses konversi bilangan biner ke bilangan heksadesimal dapat dilakukan dengan 2 cara, yaitu secara langsung dan secara tidak langsung. Proses konversi bilangan biner ke bilangan heksadesimal secara langsung dapat dilakukan dengan menggunakan algoritma berikut :

1. Jika jumlah digit bilangan biner bukan kelipatan 4, maka tambahkan bilangan 0 di depan bilangan biner hingga jumlah digit merupakan kelipatan 4.
2. Pisahkan bilangan biner ke dalam bentuk kelompok empatan.
3. Konversi masing – masing kelompok empatan tersebut ke dalam bilangan heksadesimal dengan menggunakan tabel sistem bilangan di atas.

Sebagai contoh, diambil bilangan biner 1100101101, maka proses konversi bilangan biner tersebut ke dalam bentuk bilangan heksadesimal adalah sebagai berikut :

1. Jumlah digit bilangan biner 1100101101 ada sebanyak 10 buah dan bukan merupakan kelipatan 4, sehingga harus ditambahkan 2 buah bilangan 0 di depan bilangan biner tersebut agar jumlah digit merupakan kelipatan 4.
   1100101101 → **00**1100101101
2. Pisahkan bilangan biner tersebut ke dalam bentuk kelompok empatan.
   001100101101 → 0011 | 0010 | 1101
3. Konversi masing – masing kelompok empatan tersebut ke dalam bilangan heksadesimal dengan menggunakan tabel sistem bilangan di atas.

   0011 | 0010 | 1101
   ↓ ↓ ↓
   3 2 D
4. Sehingga bilangan heksadesimal yang didapat adalah 32D.

Proses pengubahan bilangan biner ke bilangan heksadesimal secara tidak langsung dapat dilakukan dengan langkah – langkah seperti berikut,

1. Konversikan bilangan biner ke dalam bentuk bilangan desimal.

2. Konversikan bilangan desimal hasil perhitungan tersebut ke dalam bentuk bilangan heksadesimal.

Untuk mengubah satu bilangan biner ke kesetaraan desimalnya, jumlahkan kesetaraan desimal masing – masing posisi 1-nya. Sebagai contoh, diambil bilangan biner 1100101101 di atas.

$1100101101 = 1 \times 2^9 + 1 \times 2^8 + 1 \times 2^5 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0$
$= 512 + 256 + 32 + 8 + 4 + 1$
$= 813$

Sedangkan, untuk mengubah satu bilangan bulat desimal ke kesetaraan heksadesimalnya, bagilah secara berurutan bilangan tersebut dengan 16 dan konversikan angka sisanya ke dalam bentuk heksadesimal dengan urutan terbalik.

813 / 16 = 50 sisa 13 → D
50 / 16 = 3 sisa 2 → 2
3 / 16 = 0 sisa 3 → 3

Sehingga bilangan heksadesimal yang didapat adalah 32D.

Konversi dari Bilangan Heksadesimal ke Bilangan Biner

Proses konversi bilangan heksadesimal ke bilangan biner juga dapat dilakukan dengan 2 cara, yaitu secara langsung dan secara tidak langsung. Proses pengubahan bilangan heksadesimal ke bilangan biner secara langsung dapat dilakukan dengan menggunakan algoritma berikut :

1. Konversikan masing – masing digit bilangan heksadesimal ke dalam bentuk bilangan biner 4 digit.
2. Gabungkan semua bilangan biner hasil konversi tersebut sesuai dengan urutan posisinya.

Sebagai contoh, diambil bilangan heksadesimal 32D, maka proses konversi bilangan heksadesimal tersebut ke dalam bentuk bilangan biner adalah sebagai berikut :

1. Konversikan masing – masing digit bilangan heksadesimal tersebut ke dalam bentuk bilangan biner 4 digit.

3 ↓ 0011 2 ↓ 0010 D ↓ 1101

2. Gabungkan semua bilangan biner hasil konversi tersebut sesuai dengan urutan posisinya.

0011 0010 1101 → 001100101101

3. Sehingga bilangan biner yang didapat adalah 001100101101.

Proses pengubahan bilangan heksadesimal ke bilangan biner secara tidak langsung dapat dilakukan dengan langkah – langkah seperti berikut :

1. Konversikan bilangan heksadesimal ke dalam bentuk bilangan desimal.
2. Konversikan bilangan desimal hasil perhitungan tersebut ke dalam bentuk bilangan biner.

Untuk mengubah satu bilangan heksadesimal ke kesetaraan desimalnya, jumlahkan kesetaraan desimal masing – masing posisinya. Sebagai contoh, diambil bilangan heksadesimal 32D di atas.

$32D = 3 \times 16^2 + 2 \times 16^1 + D \times 16^0$
$= 768 + 32 + 13$
$= 813$

Sedangkan, untuk mengubah satu bilangan bulat desimal ke kesetaraan binernya, bagilah secara berurutan bilangan tersebut dengan 2 dan catatlah angka sisanya dengan urutan terbalik.

813 / 2 = 406 sisa 1 → 1
406 / 2 = 203 sisa 0 → 0
203 / 2 = 101 sisa 1 → 1
101 / 2 = 50 sisa 1 → 1
50 / 2 = 25 sisa 0 → 0
25 / 2 = 12 sisa 1 → 1
12 / 2 = 6 sisa 0 → 0
6 / 2 = 3 sisa 0 → 0
3 / 2 = 1 sisa 1 → 1
1/2 = 0 sisa 1 → 1

Sehingga bilangan biner yang didapat adalah 1100101101

4. WAKE (Word Auto Key Encryption)

Metode WAKE merupakan salah satu algoritma *stream cipher* yang telah digunakan secara komersial. WAKE merupakan singkatan dari *Word Auto Key Encryption*. Metode ini ditemukan oleh David Wheeler pada tahun 1993.

Metode WAKE menggunakan kunci 128 bit dan sebuah tabel 256 x 32 bit. Dalam algoritmanya, metode ini menggunakan operasi XOR, AND, OR dan *Shift Right*. Metode WAKE ini telah digunakan pada *program* Dr. Solomon Anti Virus versi terbaru.

Proses utama WAKE terdiri dari :

1. Proses pembentukan tabel *S-Box* (*Substitution Box*).
2. Proses pembentukan kunci.
3. Proses enkripsi dan dekripsi.

Inti dari metode WAKE terletak pada proses pembentukan tabel *S-Box* dan proses pembentukan kunci. Tabel *S-Box* dari metode WAKE bersifat fleksibel dan berbeda-beda untuk setiap putaran.

a. Pembentukan Tabel S-Box

Proses pembentukan tabel *S-Box* adalah sebagai berikut : [5]

1. Inisialisasi nilai TT[0] … TT[7] :
   TT[0] : 726a8f3b (dalam heksadesimal)
   TT[1] : e69a3b5c (dalam heksadesimal)
   TT[2] : d3c71fe5 (dalam heksadesimal)
   TT[3] : ab3c73d2 (dalam heksadesimal)
   TT[4] : 4d3a8eb3 (dalam heksadesimal)
   TT[5] : 0396d6e8 (dalam heksadesimal)
   TT[6] : 3d4c2f7a (dalam heksadesimal)
   TT[7] : 9ee27cf3 (dalam heksadesimal)
2. Inisialisasi nilai awal untuk T[0] … T[3] :
   T[0] = K[0] T[2] = K[2]
   T[1] = K[1] T[3] = K[3]

   K[0], K[1], K[2], K[3] dihasilkan dari kunci yang dipecah menjadi 4 bagian yang sama panjang.
3. Untuk T[4] sampai T[255], lakukan proses berikut :
   X = T[n-4] + T[n-1]
   T[n] = X >> 3 XOR TT(X AND 7)
4. Untuk T[0] sampai T[22], lakukan proses berikut :
   T[n] = T[n] + T[n+89]
5. Set nilai untuk beberapa variabel di bawah ini :
   X = T[33]
   Z = T[59] OR (01000001h)
   Z = Z AND (FF7FFFFFh)
   X = (X AND FF7FFFFFh) + Z
6. Untuk T[0] … T[255], lakukan proses berikut :
   X = (X AND FF7FFFFFh) + Z
   T[n] = T[n] AND 00FFFFFFh XOR X
7. Inisialisasi nilai untuk beberapa variabel berikut ini :
   T[256] = T[0]
   X = X AND 255
8. Untuk T[0] … T[255], lakukan proses berikut :
   Temp = (T[n XOR X] XOR X) AND 255
   T[n] = T[Temp]
   T[X] = T[n+1]

b. Pembentukan Kunci

Proses pembentukan kunci dari metode WAKE dapat ditentukan sendiri yaitu sebanyak n putaran. Semakin banyak putaran dari proses pembentukan kunci, maka keamanan datanya akan semakin terjamin. Fungsi yang digunakan dalam proses pembentukan kunci adalah M(X, Y) = (X + Y) >> 8 XOR T[(X + Y) AND 255].

Pertama-tama, kunci yang di-*input* akan dipecah menjadi 4 bagian dan di-set sebagai nilai awal dari variabel A_0, B_0, C_0, dan D_0. Nilai dari variabel ini akan diproses dengan melalui langkah berikut : [5]

$A_{i+1} = M(A_i, D_i)$

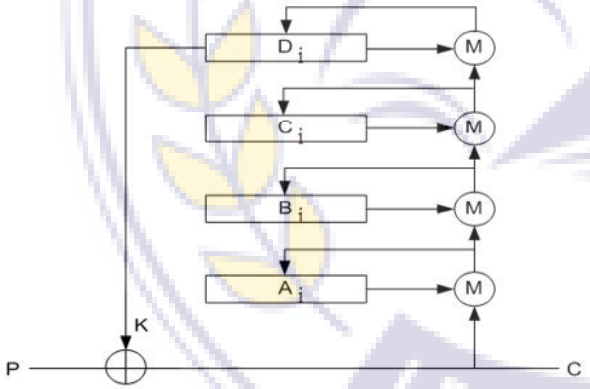
$B_{i+1} = M(B_i, A_{i+1})$

$C_{i+1} = M(C_i, B_{i+1})$

$D_{i+1} = M(D_i, C_{i+1})$

Nilai dari D_i merupakan nilai dari kunci K_i.

Agar lebih jelas, lihatlah bagan proses pembentukan kunci berikut :



Gambar 3. Bagan proses pembentukan kunci

Keterangan :

P = *Plaintext*
K = *Key*
C = *Ciphertext*
M = Fungsi M
i = Dimulai dari 0 sampai n.
Ai = Bagian pertama dari pecahan kunci
Bi = Bagian kedua dari pecahan kunci
Ci = Bagian ketiga dari pecahan kunci
Di = Bagian keempat dari pecahan kunci

d. Enkripsi dan Dekripsi

Inti dari metode WAKE tidak terletak pada proses enkripsi dan dekripsinya, karena proses enkripsi dan dekripsinya hanya berupa operasi XOR dari *plaintext* dan kunci untuk menghasilkan *ciphertext* atau operasi XOR *ciphertext* dan kunci untuk menghasilkan *plaintext*. [5]

$P = C \oplus K$

$C = P \oplus K$

dengan :

P = *Plaintext*
K = *Key*
C = *Ciphertext*

e. Perangkat Lunak Pembelajaran

Seiring dengan perkembangan peradaban manusia dan kemajuan pesat di bidang teknologi, tanpa disadari komputer telah ikut berperan dalam dunia pendidikan terutama penggunaannya sebagai alat bantu pengajaran. Percobaan penggunaan komputer untuk proses belajar dimulai di Amerika Serikat pada akhir tahun 1950-an dan awal tahun 1960-an. Kemudian penelitian selanjutnya dilakukan oleh Harvard University bekerja sama dengan IBM pada tahun 1965. Setelah munculnya komputer mikro, sistem pengajaran dengan komputer menjadi semakin meluas pada pengembangan aplikasi perangkat lunak ajar yang dikenal dengan istilah perangkat lunak pembelajaran. Perangkat lunak pembelajaran dengan komputer muncul dari sejumlah disiplin ilmu, terutama ilmu komputer dan psikologi. Dari ilmu komputer dan matematika muncul program-program yang membuat semua perhitungan dan fungsi lebih mudah dan bermanfaat. Sedangkan dari ilmu psikologi muncul pengetahuan mengenai teori belajar, teknik belajar, serta motivasi yang baik.

Banyak istilah yang dipakai untuk menyatakan perangkat lunak pembelajaran dengan komputer, seperti *Computer Assisted*

Instruction (CAI), *Computer Based Instruction* (CBI), *Computer Based Education* (CBE), *Computer Assisted Learning* (CAL), atau *Computer Based Training* (CBT).

PEMBAHASAN

Inti dari metode WAKE terletak pada proses pembentukan tabel S-Box dan proses pembentukan kunci. Proses enkripsi dan dekripsi hanya berupa operasi XOR dari plaintext dan kunci untuk menghasilkan ciphertext dan operasi XOR dari ciphertext dan kunci untuk menghasilkan plaintext.

1. Proses Pembentukan Tabel S-Box

Proses pembentukan tabel S-Box terdiri atas 8 (delapan) proses utama. Dalam prosesnya, pembentukan tabel S-Box memerlukan input kunci dengan panjang 128 bit biner atau 16 karakter ascii. Untuk lebih jelas, proses ini dapat dilihat pada contoh berikut ini.

Misalkan input key = 'WAKE, ABDUL RAHIM', maka proses pembentukan tabel S-Box dalam heksadesimal adalah sebagai berikut :

1. Inisialisasi nilai TT[0] ... TT[7].
   TT[0] = 726A8F3B (dalam heksadesimal)
   TT[1] = E69A3B5C
   TT[2] = D3C71FE5
   TT[3] = AB3C73D2
   TT[4] = 4D3A8EB3
   TT[5] = 0396D6E8
   TT[6] = 3D4C2F7A
   TT[7] = 9EE27CF3

2. Pecah kunci menjadi 4 kelompok dan masukkan pada T[0] ... T[3].
   Kunci : 'WAKE ABDUL RAHIM'
   Kode ascii dari 'W' = 87 = 57
   Kode ascii dari 'A' = 65 = 41
   Kode ascii dari 'K' = 75 = 4B
   Kode ascii dari 'E' = 69 = 45
   Kode ascii dari ' ' = 32 = 20
   Kode ascii dari 'A' = 65 = 41
   Kode ascii dari 'B' = 66 = 42
   Kode ascii dari 'D' = 68 = 44
   Kode ascii dari 'U' = 85 = 55
   Kode ascii dari 'L' = 76 = 4C
   Kode ascii dari ' ' = 32 = 20
   Kode ascii dari 'R' = 82 = 52
   Kode ascii dari 'A' = 65 = 41
   Kode ascii dari 'H' = 72 = 48
   Kode ascii dari 'I' = 73 = 49
   Kode ascii dari 'M' = 77 = 4D
   Kunci (dalam heksa) = 57414B4520414244554C20524148494D
   T[0] = K[0] = 57414B45
   T[1] = K[1] = 20414244
   T[2] = K[2] = 554C2052
   T[3] = K[3] = 4148494D

3. Untuk n = 4 sampai 255, lakukan prosedur berikut :

   X = T[n-4] + T[n-1]
   T[n] = X>>3 XOR TT(X AND 7)

   n = 4
   -> X = T[0] + T[3] = 57414B45 + 4148494D = 98899492
   -> X >> 3 (Shift Right 3 bit) = 98899492 >> 3 = 13113292
   X AND 7 = 98899492 AND 7(10) = 2
   T[4] = X >> 3 XOR TT[X AND 7] = 13113292 XOR TT[2] = C0D62D77
   n = 5
   -> X = T[1] + T[4] = 20414244 + C0D62D77 = E1176FBB
   -> X >> 3 (Shift Right 3 bit) = E1176FBB >> 3 = 1C22EDF7
   X AND 7 = E1176FBB AND 7(10) = 3
   T[5] = X >> 3 XOR TT[X AND 7] = 1C22EDF7 XOR TT[3] = B71E9E25
   n = 6
   -> X = T[2] + T[5] = 554C2052 + B71E9E25 = 0C6ABE77

-> X >> 3 (Shift Right 3 bit) = 0C6ABE77 >> 3 = 018D57CE
X AND 7 = 0C6ABE77 AND 7(10) = 7
T[6] = X >> 3 XOR TT[X AND 7] = 018D57CE XOR TT[7] = 9F6F2B3D (dan seterusnya hingga n = 255).

4. Untuk n = 0 sampai 22, lakukan prosedur berikut :

T[n] = T[n] + T[n + 89]

n = 0
T[0] = T[0] + T[89] = 57414B45 + 8B7FC84C = E2C11391
n = 1
T[1] = T[1] + T[90] = 20414244 + 2A40998C = 4A81DBD0
n = 2
T[2] = T[2] + T[91] = 554C2052 + 13B25123 = 68FE7175
n = 3
T[3] = T[3] + T[92] = 4148494D + DB508746 = 1C98D093
(dan seterusnya hingga n = 22).

5. Set nilai untuk beberapa variabel di bawah ini.
X = B860D0A5
Z = T[59] OR 01000001 = D0EA526D OR 01000001 = D1EA526D
Z = Z AND FF7FFFFF = D1EA526D AND FF7FFFFF = D16A526D
X = X AND FF7FFFFF = B860D0A5 AND FF7FFFFF = 89CB2312.

6. Untuk n = 0 sampai 255, lakukan prosedur berikut :

X = (X AND FF7FFFFF) + Z
T[n] = T[n] AND 00FFFFFF XOR X

n = 0
X = (89CB2312 AND FF7FFFFF) + D16A526D = 5AB5757F
T[0] = E2C11391] AND 00FFFFFF XOR 5AB5757F = 5A7466EE
n = 1
X = (5AB5757F AND FF7FFFFF) + D16A526D = 2B9FC7EC
T[1] = 4A81DBD0] AND 00FFFFFF XOR 2B9FC7EC = 2B1E1C3C

n = 2
X = (2B9FC7EC AND FF7FFFFF) + D16A526D = FC8A1A59
T[2] = 68FE7175] AND 00FFFFFF XOR FC8A1A59 = FC746B2C
(dan seterusnya hingga n = 255).

7. Set nilai untuk beberapa variabel berikut.
T[256] = T[0] = 5A7466EE
X = X AND 255(10) = 899D9012 AND 255(10) = 00000012

8. Untuk n = 0 sampai 255, lakukan prosedur berikut.

Temp = (T[n XOR X] XOR X) AND 255
T[n] = T[Temp]
T[X] = T[n+1]

n = 0
Temp = T[18] XOR X AND 255 = 0CE96C25 XOR 00000012 AND 255 = 00000037
T[0] = T[55] = 4113C1FC
T[18] = T[1] = 2B1E1C3C
n = 1
Temp = T[19] XOR X AND 255 = DDC78A60 XOR 00000012 AND 255 = 00000072
T[1] = T[114] = 6C9B0C39
T[18] = T[2] = FC746B2C
n = 2
Temp = T[16] XOR X AND 255 = 6A5BD750 XOR 00000012 AND 255 = 00000042
T[2] = T[66] = 3CF29ED4
T[18] = T[3] = CDECBC55

n = 3
Temp = T[17] XOR X AND 255 = 3B383E49 XOR 00000012 AND 255 = 0000005B
T[3] = T[91] = A532931D
T[18] = T[4] = 9E2DFCBD
(dan seterusnya hingga n = 255).

2. Proses Pembentukan Kunci

Proses pembentukan kunci memerlukan *input kunci* dengan panjang 128 bit biner atau 16 karakter *ascii*. Pertama – tama, *input* kunci dipecah menjadi 4 kelompok dan di-*set* sebagai nilai awal dari variabel A_0, B_0, C_0, D_0. Kemudian isi variabel A, B, C dan D dan ulangi sebanyak n-putaran yang di-*input*.

$A_{i+1} = M(A_i, D_i)$
$B_{i+1} = M(B_i, A_{i+1})$
$C_{i+1} = M(C_i, B_{i+1})$
$D_{i+1} = M(D_i, C_{i+1})$

Fungsi M(X, Y) = (X + Y)>>8 XOR T[(X + Y) AND 255]. Nilai dari D_i merupakan nilai dari kunci K_i. Proses ini dapat dilihat pada contoh berikut :
Misalkan *input key* : 'WAKE ABDUL RAHIM' dan putaran kunci sebanyak 5 kali, maka proses pembentukan kunci dalam heksadesimal adalah sebagai berikut:
Kunci 'WAKE ABDUL RAHIM' diubah dalam bentuk heksa =
57414B4520414244554C20524148494D

Pecah kunci menjadi 4 kelompok dan masukkan ke A(0), B(0), C(0) dan D(0).
A(0) = 57414B45
B(0) = 20414244
C(0) = 554C2052
D(0) = 4148494D

KUNCI PUTARAN 1

FungsiM(A[0],D[0]) =
FungsiM(57414B45,4148494D) = (57414B45 + 4148494D)>>8 XOR T[(57414B45 + 4148494D) AND 255(10)] = 98899492>>8 XOR T[146] = 00988994 XOR 587FC408 = 58E74D9C
A[1] = 58E74D9C

FungsiM(B[0],A[1]) =
FungsiM(20414244,58E74D9C) = (20414244 + 58E74D9C)>>8 XOR T[(20414244 + 58E74D9C) AND 255(10)] = 79288FE0>>8 XOR T[224] = 0079288F XOR 562D1761 = 56543FEE
B[1] = 56543FEE

FungsiM(C[0],B[1]) =
FungsiM(554C2052,56543FEE) = (554C2052 + 56543FEE)>>8 XOR T[(554C2052 + 56543FEE) AND 255(10)] = ABA06040>>8 XOR T[64] = 00ABA060 XOR 45A69C26 = 450D3C46
C[1] = 450D3C46

FungsiM(D[0],C[1]) =
FungsiM(4148494D,450D3C46) = (4148494D + 450D3C46)>>8 XOR T[(4148494D + 450D3C46) AND 255(10)] = 86558593>>8 XOR T[147] = 00865585 XOR 587FC408 = 58F9918D
D[1] = 58F9918D.

KUNCI PUTARAN 2

FungsiM(A[1],D[1]) =
FungsiM(58E74D9C,58F9918D) = (58E74D9C + 58F9918D)>>8 XOR T[(58E74D9C + 58F9918D) AND 255(10)] = B1E0DF29>>8 XOR T[41] = 00B1E0DF XOR 0B9DD27C = 0B2C32A3
A[2] = 0B2C32A3

FungsiM(B[1],A[2]) =
FungsiM(56543FEE,0B2C32A3) = (56543FEE + 0B2C32A3)>>8 XOR T[(56543FEE + 0B2C32A3)

AND 255(10)] = 61807291>>8 XOR T[145] = 00618072 XOR 4113C1FC = 4172418E
B[2] = 4172418E

FungsiM(C[1],B[2]) = FungsiM(450D3C46,4172418E) = (450D3C46 + 4172418E)>>8 XOR T[(450D3C46 + 4172418E) AND 255(10)] = 867F7DD4>>8 XOR T[212] = 00867F7D XOR 0B9DD27C = 0B1BAD01
C[2] = 0B1BAD01

FungsiM(D[1],C[2]) = FungsiM(58F9918D,0B1BAD01) = (58F9918D + 0B1BAD01)>>8 XOR T[(58F9918D + 0B1BAD01) AND 255(10)] = 64153E8E>>8 XOR T[142] = 0064153E XOR DB1F1E80 = DB7B0BBED[2] = DB7B0BBE

KUNCI PUTARAN 3

FungsiM(A[2],D[2]) = FungsiM(0B2C32A3,DB7B0BBE) = (0B2C32A3 + DB7B0BBE)>>8 XOR T[(0B2C32A3 + DB7B0BBE) AND 255(10)] = E6A73E61>>8 XOR T[97] = 00E6A73E XOR 9740616C = 97A6C652
A[3] = 97A6C652

FungsiM(B[2],A[3]) = FungsiM(4172418E,97A6C652) = (4172418E + 97A6C652)>>8 XOR T[(4172418E + 97A6C652) AND 255(10)] = D91907E0>>8 XOR T[224] = 00D91907 XOR 562D1761 = 56F40E66
B[3] = 56F40E66

FungsiM(C[2],B[3]) = FungsiM(0B1BAD01,56F40E66) = (0B1BAD01 + 56F40E66)>>8 XOR T[(0B1BAD01 + 56F40E66) AND 255(10)] = 620FBB67>>8 XOR T[103] = 00620FBB XOR 45A69C26 = 45C4939D
C[3] = 45C4939D

FungsiM(D[2],C[3]) = FungsiM(DB7B0BBE,45C4939D) = (DB7B0BBE + 45C4939D)>>8 XOR T[(DB7B0BBE + 45C4939D) AND 255(10)] = 213F9F5B>>8 XOR T[91] = 00213F9F XOR ACAC0250 = AC8D3DCF
D[3] = AC8D3DCF

KUNCI PUTARAN 4

FungsiM(A[3],D[3]) = FungsiM(97A6C652,AC8D3DCF) = (97A6C652 + AC8D3DCF)>>8 XOR T[(97A6C652 + AC8D3DCF) AND 255(10)] = 44340421>>8 XOR T[33] = 00443404 XOR 562D1761 = 56692365
A[4] = 56692365

FungsiM(B[3],A[4]) = FungsiM(56F40E66,56692365) = (56F40E66 + 56692365)>>8 XOR T[(56F40E66 + 56692365) AND 255(10)] = AD5D31CB>>8 XOR T[203] = 00AD5D31 XOR D9095067 = D9A40D56
B[4] = D9A40D56

FungsiM(C[3],B[4]) = FungsiM(45C4939D,D9A40D56) = (45C4939D + D9A40D56)>>8 XOR T[(45C4939D + D9A40D56) AND 255(10)] = 1F68A0F3>>8 XOR T[243] = 001F68A0 XOR ACAC0250 = ACB36AF0
C[4] = ACB36AF0

FungsiM(D[3],C[4]) = FungsiM(AC8D3DCF,ACB36AF0) = (AC8D3DCF + ACB36AF0)>>8 XOR T[(AC8D3DCF + ACB36AF0) AND 255(10)] = 5940A8BF>>8 XOR T[191] = 005940A8 XOR 001A0B07 = 00434BAFD[4] = 00434BAF.

KUNCI PUTARAN 5

FungsiM(A[4],D[4]) = FungsiM(56692365,00434BAF) = (56692365 + 00434BAF)>>8 XOR T[(56692365 + 00434BAF) AND 255(10)] = 56AC6F14>>8 XOR T[20] = 0056AC6F XOR EDC8EE67 = ED9E4208
A[5] = ED9E4208

FungsiM(B[4],A[5]) =
FungsiM(D9A40D56,ED9E4208) = (D9A40D56 + ED9E4208)>>8 XOR T[(D9A40D56 + ED9E4208) AND 255(10)] = C7424F5E>>8 XOR T[94] = 00C7424F XOR 8C1FDB59 = 8CD89916
B[5] = 8CD89916

FungsiM(C[4],B[5]) =
FungsiM(ACB36AF0,8CD89916) = (ACB36AF0 + 8CD89916)>>8 XOR T[(ACB36AF0 + 8CD89916) AND 255(10)] = 398C0406>>8 XOR T[6] = 00398C04 XOR B7354D31 = B70CC135
C[5] = B70CC135

FungsiM(D[4],C[5]) =
FungsiM(00434BAF,B70CC135) = (00434BAF + B70CC135)>>8 XOR T[(00434BAF + B70CC135) AND 255(10)] = B7500CE4>>8 XOR T[228] = 00B7500C XOR 7940364E = 79F76642
D[5] = 79F76642
KUNCI = D[5] = 79F76642.

3. Proses Enkripsi

Proses enkripsi dari metode WAKE untuk menghasilkan *ciphertext* adalah berupa hasil operasi XOR dari *plaintext* dan 32 bit kunci yang dihasilkan dari proses pembentukan kunci.

***Ciphertext* (C) = *Plaintext* (P) XOR *Key* (K)**

Plaintext : 'COBA'
Kode ascii dari 'C' = 43
Kode ascii dari 'O' = 4F
Kode ascii dari 'B' = 42
Kode ascii dari 'A' = 41
Plain Text (dalam heksa) = 434F4241
Kunci dari proses pembentukan kunci = 79F76642
Ciphertext = Plaintext XOR Key
43 XOR 79 = 3A = ':'
4F XOR F7 = B8 = '¸'
42 XOR 66 = 24 = '$'
41 XOR 42 = 03 = '□'
Hasil proses enkripsi = :¸$□

4. Proses Dekripsi

Proses dekripsi dari metode WAKE untuk menghasilkan *plaintext* adalah berupa hasil operasi XOR dari *ciphertext* dan 32 bit kunci yang dihasilkan dari proses pembentukan kunci.

***Plaintext* (P) = *Ciphertext* (C) XOR *Key* (K)**

Ciphertext : ' :¸$□ '
Kode ascii dari ':' = 3A
Kode ascii dari '¸' = B8
Kode ascii dari '$' = 24
Kode ascii dari '□' = 03
Ciphertext (dalam heksa) = 3AB82403
Kunci dari proses pembentukan kunci = 79F76642
Plaintext = Ciphertext XOR Key
3A XOR 79 = 43 = 'C'
B8 XOR F7 = 4F = 'O'
24 XOR 66 = 42 = 'B'
03 XOR 42 = 41 = 'A'
Hasil proses dekripsi = COBA.

5. Algoritma Pembentukan Tabel S-Box

Proses pembentukan tabel *S-Box* memerlukan *input* kunci 16 karakter *ascii* atau 128 bit biner, sehingga tabel *S-Box* pada metode WAKE adalah berbeda untuk *input* kunci yang berbeda.

Algoritma proses pembentukan tabel *S-Box* adalah sebagai berikut,
{1. Inisialisasi nilai TT[0] - T[7]}
TT(0) = FHexToBiner("726a8f3b")
TT(1) = FHexToBiner("e69a3b5c")
TT(2) = FHexToBiner("d3c71fe5")
TT(3) = FHexToBiner("ab3c73d2")
TT(4) = FHexToBiner("4d3a8eb3")
TT(5) = FHexToBiner("0396d6e8")
TT(6) = FHexToBiner("3d4c2f7a")
TT(7) = FHexToBiner("9ee27cf3")

{2. Pecah kunci (128 bit) menjadi 4 kelompok dan masukkan nilainya ke T[0], T[1], T[2], T[3]}
X = ""

```
For N = 1 To Len(pcKunci)
   X = X &
FormatS(FDecToBiner(Asc(Mid(pcKunci,N,
1))),"0",8)
Next N
T(0) = Mid(X, 1, 32)
T(1) = Mid(X, 33, 32)
T(2) = Mid(X, 65, 32)
T(3) = Mid(X, 97, 32)

{3. Untuk N = 4 sampai 255, lakukan proses
berikut}
For N = 4 To 255
   {X = T[n-4] + T[n-1]}
   X = FAddBiner(T(N - 4), T(N - 1), 32)
   {T[n] = X>>3 XOR TT[X AND 7]}
   T(N) = FOpBiner("XOR", FShiftRight(X, 3), _
   TT(FBinerToDec(FOpBiner("AND", X,
"111"))), 32)
Next N

{4. Untuk N = 0 sampai 22, lakukan proses
berikut}
For N = 0 To 22
   {T[n] = T[n] + T[n+89]}
   T(N) = FAddBiner(T(N), T(N + 89), 32)
Next N

{5. Set nilai untuk variabel di bawah ini}
X = T(33)
Z = FOpBiner("OR", T(59),
FHexToBiner("01000001"), 32)
Z = FOpBiner("AND", Z,
FHexToBiner("FF7FFFFF"), 32)
X = FAddBiner(FOpBiner("AND", X,
FHexToBiner("FF7FFFFF"),32), Z, 32)

{6. Untuk N = 0 sampai 255, lakukan proses
berikut}
For N = 0 To 255
   {X = (X And FF7FFFFF) + Z}
   X = FAddBiner(FOpBiner("AND",X,_
FHexToBiner("FF7FFFFF"),32),Z,32)
   {T[n] = T[n] AND 00FFFFFF XOR X}
   T(N) = FOpBiner("XOR", FOpBiner("AND",
T(N), _
                FHexToBiner("00FFFFFF"),
                32), X, 32)
Next N

{7. Inisialisasi nilai untuk beberapa variabel
berikut}
T(256) = T(0)
X = FOpBiner("AND", X, FDecToBiner(255), 32)

{8. Untuk N = 0 sampai T[255], lakukan proses
berikut}
For N = 0 To 255
   {Temp = (T[n XOR X] XOR X) AND 255}
   Temp = T(FBinerToDec(FOpBiner("XOR",
FDecToBiner(N), X, 32)))
   Temp = FOpBiner("XOR", Temp, X, 32)
   Temp = FOpBiner("AND", Temp,
FDecToBiner(255), 32)
   {T[n] = T[Temp]}
   T(N) = T(FBinerToDec(Temp))
   {T[X] = T[n+1]}
   T(FBinerToDec(X)) = T(N + 1)
Next N
```

6. Algoritma Pembentukan Kunci

Proses pembentukan kunci pada metode WAKE adalah sebanyak n putaran. Banyak putaran (besar n) ditentukan oleh *user*. Hasil pembentukan kunci berbeda untuk setiap putaran.

Algoritma proses pembentukan kunci adalah sebagai berikut,

```
{1. Ubah kunci 128 bit ke biner, pecah menjadi
4 kelompok dan masukkan nilainya ke A[0],
B[0], C[0], D[0] masing – masing 32 bit}
X = ""
For N = 1 To Len(pcKunci)
   X = X &
FormatS(FDecToBiner(Asc(Mid(pcKunci,N,
1))),"0",8)
Next N
A(0) = Mid(X, 1, 32)
```

```
B(0) = Mid(X, 33, 32)
C(0) = Mid(X, 65, 32)
D(0) = Mid(X, 97, 32)
{Putaran kunci – n putaran}
For N = 1 To pnPutaran
   A(N) = FungsiM(A(N - 1), D(N - 1))
   B(N) = FungsiM(B(N - 1), A(N))
   C(N) = FungsiM(C(N - 1), B(N))
   D(N) = FungsiM(D(N - 1), C(N))
Next N
{Kunci yang terbentuk  =  D}
strKunciBiner = D(N - 1)
```

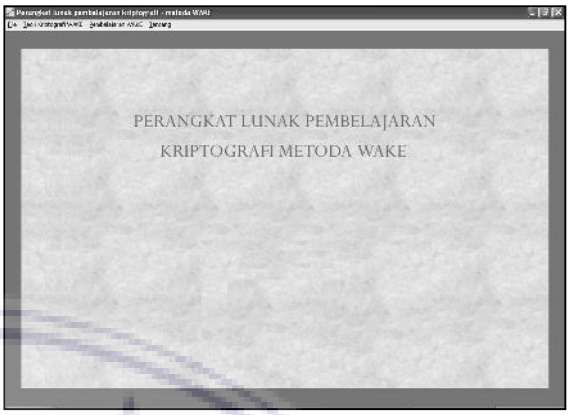
Penulis merancang algoritma untuk fungsi M(X,Y) yang dipakai di dalam proses ini dalam bentuk sebuah fungsi. Algoritma fungsi M(X,Y) adalah sebagai berikut,

```
{FUNGSI M(X,Y)}
Public Function FungsiM(pX As String, pY As
String) As String
   Dim Temp As String
   Dim Temp1 As String
   {Temp1 = X + Y}
   Temp1 = FAddBiner(pX, pY, 32)
  {Temp = Temp1 Shift >> 8 kali}
   Temp = FShiftRight(Temp1, 8)
  {Temp1 = (X + Y) And 255}
   Temp1 = FBinerToDec(FOpBiner("AND",
Temp1, "11111111"))
   {Temp1 = T[(X + Y) And 255)]}
   Temp1 = T(Val(Temp1))
   {di-XOR T(X + Y)}
   Temp = FOpBiner("XOR", Temp, Temp1, 32)
  {Kembalikan nilai ke fungsi M}
   FungsiM = Temp
End Function
```

7. Perangkat Lunak

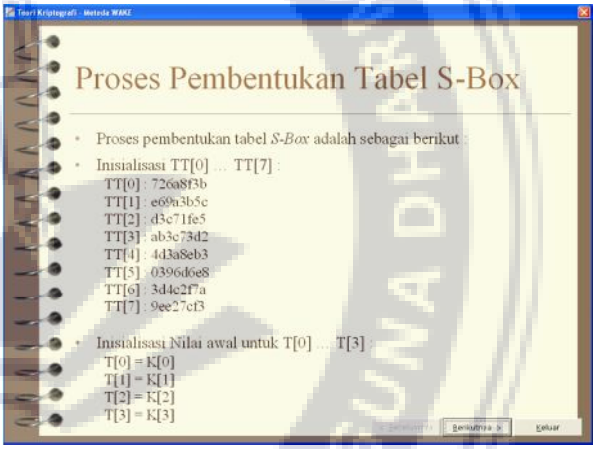
a. Tampilan Output

Klik pada file '*Wake-Crypto.exe*' untuk menjalankan perangkat lunak. Setelah itu akan muncul tampilan *form* '*Main*'. *Form* '*Main*' memiliki menu - menu yang digunakan untuk memanggil *form* sesuai dengan fungsi perangkat lunak pembelajaran.



Gambar 4. Tampilan Form Main

Apabila menu [Teori Kriptografi WAKE] di-klik dan salah satu sub menu teori dipilih, maka akan muncul *form* teori yang berisi teori singkat mengenai kriptografi metode WAKE.

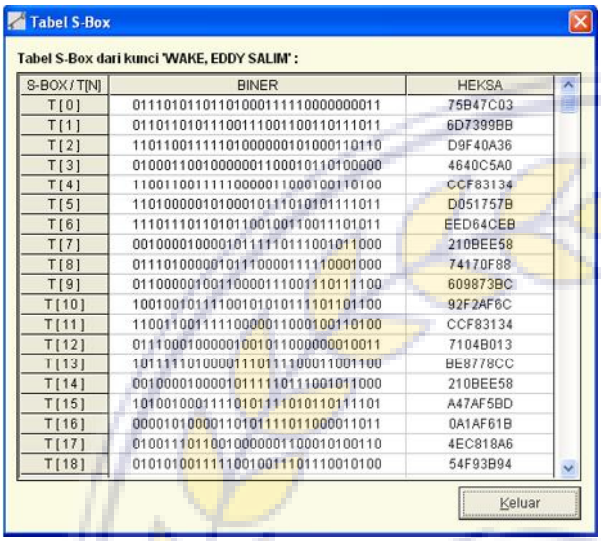


Gambar 5. Tampilan Form Teori

Untuk melihat proses pembentukan tabel *S-Box*, pilih menu 'Pembelajaran WAKE' dan klik sub menu 'Proses S-Box'. Muncul *form input* berikut.

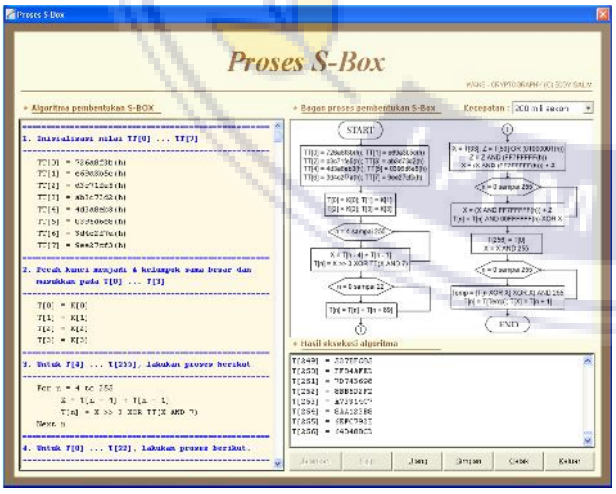
Gambar 6. Tampilan Form Input Proses S-Box

Untuk melihat hasil dari pembentukan tabel *S-Box* tanpa mengikuti proses secara bertahap, klik pada tulisan biru 'Lihat Hasil / Tabel S-Box' di sebelah kiri bawah *form*, maka akan muncul *form* berikut.



Gambar 7. Tampilan Form Tabel S-Box

Apabila *user* ingin melihat dan mengikuti prosedur kerja proses pembentukan tabel *S-Box* secara bertahap, maka klik pada tombol 'Proses kerja S-Box'. Selanjutnya, muncul *form* 'Proses S-Box'.

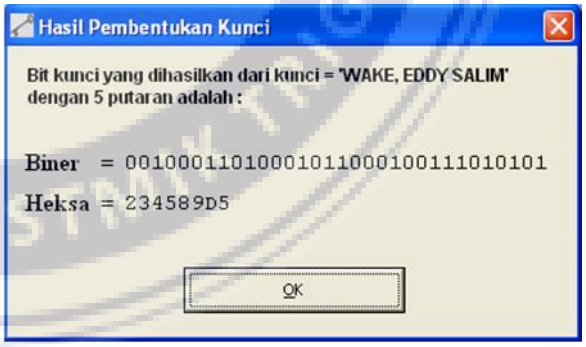


Gambar 8. Tampilan Form Proses S-Box

Untuk melihat proses pembentukan kunci, pilih menu 'Pembelajaran WAKE' dan klik sub menu 'Proses Pembentukan Kunci'. Muncul *form input* berikut.

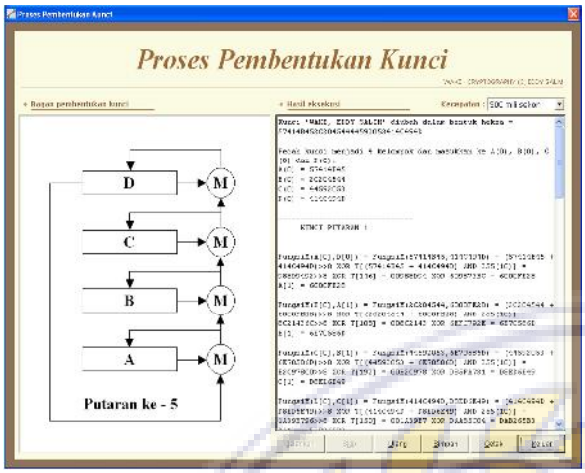
Gambar 9. Form Input Proses Pembentukan Kunci

Klik pada tulisan biru 'Lihat Tabel S-Box' di sebelah kiri bawah form akan memunculkan form 'Tabel S-Box' pada gambar 4.4. Untuk melihat hasil pembentukan kunci tanpa mengikuti proses secara bertahap, klik pada tulisan biru 'Kunci yang terbentuk' di sebelah kiri bawah form, maka akan muncul form berikut.



Gambar 10. Form Hasil Pembentukan Kunci

Apabila user ingin melihat dan mengikuti prosedur kerja proses pembentukan kunci secara bertahap, maka klik pada tombol 'Proses'. Selanjutnya, muncul form 'Proses Pembentukan Kunci'.



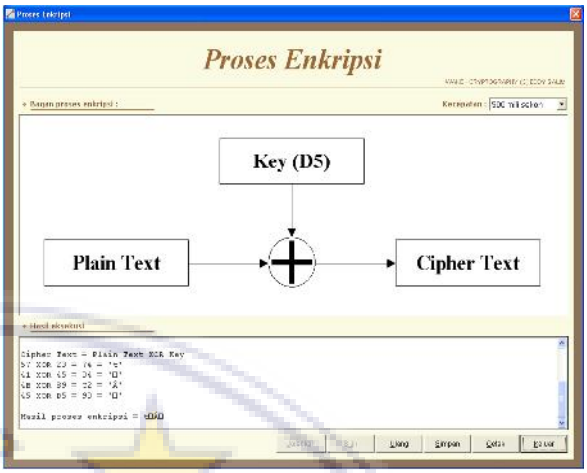
Gambar 11. Form Proses Pembentukan Kunci

Untuk melihat proses enkripsi, pilih menu ‘Pembelajaran WAKE’ dan klik sub menu ‘Proses Enkripsi’. Muncul form input berikut.

Gambar 12. Form Input Proses Enkripsi

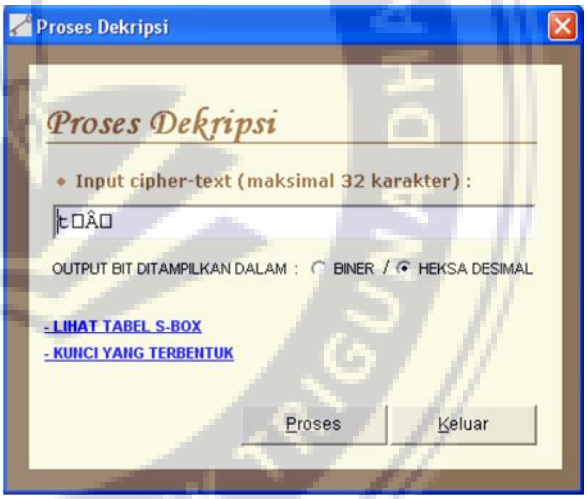
Klik pada tulisan biru ‘Lihat Tabel S-Box’ di sebelah kiri bawah *form* akan memunculkan *form* ‘Tabel S-Box’ pada gambar 12, sedangkan klik pada tulisan biru ‘Kunci yang terbentuk’ akan memunculkan *form* ‘Hasil Pembentukan Kunci’ pada gambar 13.

Untuk melihat dan mengikuti prosedur kerja proses enkripsi secara bertahap, maka klik pada tombol ‘Proses’. Selanjutnya, muncul *form* ‘Proses Enkripsi’.



Gambar 13. Form Proses Enkripsi

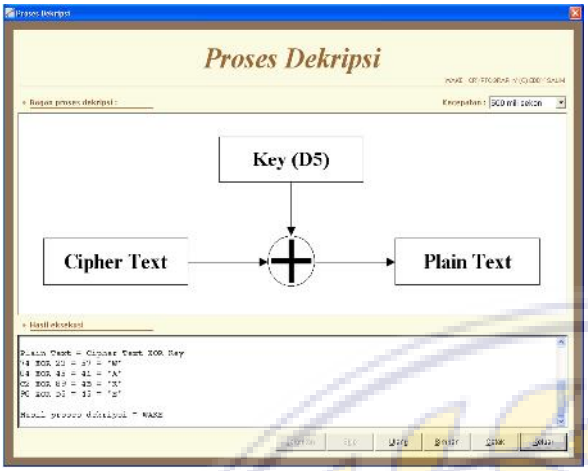
Untuk melihat proses dekripsi, pilih menu ‘Pembelajaran WAKE’ dan klik sub menu ‘Proses Dekripsi’. Muncul *form input* berikut.



Gambar 14. Form Input Proses Dekripsi

Klik pada tulisan biru ‘Lihat Tabel S-Box’ di sebelah kiri bawah form akan memunculkan form ‘Tabel S-Box’ pada gambar 14, sedangkan klik pada tulisan biru ‘Kunci yang terbentuk’ akan memunculkan form ‘Hasil Pembentukan Kunci’ pada gambar 15.

Untuk melihat dan mengikuti prosedur kerja proses dekripsi secara bertahap, maka klik pada tombol ‘Proses’. Selanjutnya, muncul form ‘Proses Dekripsi’.



Gambar 15. Form Proses Dekripsi

SIMPULAN

Setelah selesai menyelesaikan perancangan perangkat lunak pembelajaran kriptografi metode WAKE ini, penulis menarik kesimpulan sebagai berikut :

1. Perangkat lunak ini menunjukkan setiap langkah dan tahapan proses – proses (proses pembentukan tabel *S-Box*, proses pembentukan kunci, proses enkripsi dan proses dekripsi) yang terdapat di dalam kriptografi metode WAKE, sehingga dapat membantu pemahaman atau pembelajaran prosedur kerja atau algoritma dari metode kriptografi tersebut.
2. Semakin banyak putaran dari proses pembentukan kunci, maka keamanan data akan semakin terjamin.
3. Kelebihan dari perangkat lunak ini dapat menggambarkan algoritma dan menunjukkan tahapan-tahapan pada proses enkripsi dan dekripsi pesan dengan metode kriptografi WAKE sehingga secara khusus memudahkan mahasiswa dalam memahami kriptografi metode WAKE dan keamanan data secara umum.

Kekurangan dari perangkat lunak ini belum dapat untuk mengenkripsi file selain file text, seperti file music (*.mp3), gambar (*.jpg), Ms. Word (*.doc) dan lain sebagainya.

DAFTAR PUSTAKA

K. Jusuf. 2002. *Kriptografi, Keamanan Internet dan Jaringan Komunikasi.* Bandung: Penerbit Informatika.

S.Ario. 2001. *Microsoft Visual Basic 6.0.* PT. Elex Media Komputindo.

S.Bruce. 1996. *Applied Crytography*. Second Edition, John Wiley & Sons, Inc.

Sodhi, Jag. 1991. *Software Engineering Methods, Management, and CASE Tools, TAB Professional dan Reference Books*. Amerika.

http://www.cix.co.uk/~klockstone/wake. htm, tanggal 11 Juli 2005.

http://www.cix.co.uk/~klockstone/ hereward.htm, tanggal 11 Juli 2005.

http://eprint.iacr.org/2001/065.pdf, tanggal 11 Juli 2005.