PENGEMBANGAN KRIPTOGRAFI KUNCI SIMETRIS "SONICS CIPHER"

Dhamma Nibbana Putra, NIM: 13503040

Departemen Informatika, Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung

e-mail: if13040@students.if.itb.ac.id

ABSTRAKSI

Makalah ini akan membahas tentang pengembangan suatu metoda enkripsi baru yang akan dinamakan *"Sonics Cipher"* untuk menyandikan sebuah pesan rahasia. *Sonics Cipher* sendiri merupakan teknik kriptografi kunci simetris yang dikembangkan sendiri dengan tujuan menjadi algoritma enkripsi yang cepat, mudah dikustomisasi (sehingga menyulitkan kriptanalisis), dapat didekripsi dengan mudah jika mengetahui kunci yang tepat, dan tentu saja mencoba menjadi *unbreakable cipher*.

Untuk pengembangan dan pengujian akan dibuat sebuah perangkat lunak bernama *Sonics D/Encryptor* dengan fitur kustomisasi yang belum ditentukan. Perangkat lunak ini akan dikembangkan dengan *Microsoft Visual Studio 2005* dan dukungan terhadap *.NET framework* yang utamanya akan berjalan pada sistem operasi *Microsoft Windows. Sonics D/Encryptor* akan menjadi *platform* pengujian dan analisis utama dalam pengembangan *Sonics Cipher.*

Untuk pengujian terhadap keamanan dari *Sonics Cipher* sendiri akan dicek dengan menggunakan tes dekripsi terhadap dokumen enkripsi yang telah diubah isinya. Tes ini bertujuan untuk melihat seberapa besar ketahanan *Sonics Cipher* terhadap serangan dari luar yang bertujuan untuk memanipulasi informasi yang dikirimkan. Sampai saat ini peforma dari *Sonics Cipher* masih sangat baik karena penginterupsian pesan baik dengan cara penyisipan, pengubahan, atau penghapusan sebagian dari pesan dapat diketahui, serta belum adanya cara yang pasti untuk memecahkan *Sonics Cipher* membuat teknik enkripsi ini masih sangat baik untuk digunakan.

Kata Kunci: *kriptografi, kriptanalisis, kunci simetris, sonics cipher.*

1. Pendahuluan

Pengiriman informasi telah lama dilakukan sepanjang sejarah manusia. Sejak jaman dahulu usaha pengiriman pesan yang tidak boleh diketahui oleh pihak lain telah banyak melahirkan cerita-cerita menarik. Sejarah sendiri telah menyaksikan beberapa peristiwa penting terjadi ketika usaha-usaha tersebut berhasil ataupun gagal.

Sampai saat ini, upaya pembuatan metode *cipher* yang tidak terpecahkan terus

dikembangkan. Dengan pesatnya kemajuan teknologi komputasi, maka pembuatan algoritma enkripsi saat ini makin didasari pada operasi matematika yang melibatkan perkalian atau perpangkatan bilangan yang sangat besar. Selain itu pula, enkripsi lebih banyak dilakukan pada rangkaian bit biner, mengingat penggunaan komputer pada saat ini yang masih memproses data dalam bentuk biner. Karena itu, algoritma enkripsi yang cukup kuat dan masih memungkinkan untuk didekripsi secara manual tanpa bantuan komputer masih menarik untuk dipelajari.

2. Kriptografi kunci simetris

Berdasarkan jenis kuncinya, kriptografi dapat dibedakan menjadi 2, yaitu kriptografi kunci simetris dan kriptografi kunci non-simetris. Pada kriptografi kunci simetris, kunci yang dipakai untuk mengenkripsi dan mendekripsi pesan sama, sedangkan pada kriptografi kunci non-simteris, kunci yang dipakai berbeda. Kriptografi kunci simetris dipilih karena proses enkripsi-dekripsinya yang relatif lebih mudah dibandingkan dengan kriptografi kunci non-simetris. Metode kriptografi kunci non-simetris yang ada saat ini biasanya memerlukan bantuan komputer untuk melakukan enkripsi sehingga tidak cocok untuk pembuatan metode kriptografi yang bertujuan agar dapat digunakan secara manual tanpa bantuan komputer.

3. Teknik kriptanalisis sederhana

3.1. Subtitusi huruf

Banyak teknik kriptografi berdasarkan penggantian huruf. Teknik yang paling terkenal adalah *Caesar Cipher* yang digunakan oleh Julius Caesar pada jaman dahulu. Pada *Caesar Cipher* prinsip penggantian huruf dapat digambarkan secara sederhana dimana setiap abjad mempunyai penggantinya masing-masing yaitu huruf ke-N sesudah huruf tersebut. Misalnya jika nilai N adalah 3 dan ingin mengganti huruf 'A', maka abjad penggantinya adalah huruf 'D', karena 'D' terletak pada tempat ketiga sesudah 'A' dalam urutan alfabet.

Meskipun sederhana dan sangat mudah dipecahkan, teknik ini memiliki keunggulan antara lain proses dekripsi yang cepat jika telah mengetahui kuncinya. Oleh karena itu, teknik subtitusi telah mengalami perkembangan, misalnya dengan menggunakan beberapa simbol untuk menggantikan suatu simbol, atau penggunaan sebuah simbol yang dipakai secara bersama-sama dengan simbol yang lain sehingga menyulitkan para kriptanalis dalam menentukan simbol pengganti yang tepat. Penggantian simbol ini biasanya diatur oleh sebuah *rules* yang bergantung pada sebuah kata kunci. *Vigenere Cipher* menggunakan teknik ini.

3.2. Analisis frekuensi

Cara lain dalam kriptanalisis adalah dengan memperhatikan frekuensi kemunculan huruf atau simbol yang berulang. Jika telah diketahui bahwa suatu pesan menggunakan substitusi abjad tunggal, maka dengan teknik analisis frekuensi, pesan tersebut dapat dipecahkan dengan mudah. Analisis frekuensi pada dasarnya adalah penghitungan jumlah kemunculan huruf pada *ciphertext* dan membandingkannya dengan frekuensi kemunculan huruf pada kata-kata yang

umum. Misalnya jika diketahui bahwa suatu pesan menggunakan bahasa Inggris dan setelah dihitung, huruf 'J' muncul paling banyak dalam pesan tersebut, maka dapat diasumsikan bahwa huruf 'J' menggantikan huruf 'E' sebab pada kalimat dalam bahasa Inggris pada umumnya, huruf 'E' merupakan huruf yang paling sering muncul. Dengan menghitung beberapa huruf yang paling sering muncul pada pesan dan dibandingkan dengan huruf-huruf yang biasanya sering muncul pada kalimat yang umum, maka sebagian dari pesan dapat diketahui dan huruf-huruf pengganti lainnya dapat ditemukan.

Pada metode kriptografi sederhana, teknik analisis frekuensi sangat berguna sebab sangat sulit untuk menghindari pemakaian kata-kata yang umum dipakai dalam kehidupan sehari-hari, seperti kata 'THE' dalam bahasa Inggris.

3.3. Analisis bentuk pesan

Untuk dapat memecahkan suatu sandi, biasanya seorang kriptanalis akan mengamati pesan yang ia terima secara keseluruhan. Beberapa teknik penyandian biasanya membuat pola tertentu yang jika diamati dengan teliti, sorang kriptanalis mungkin saja dapat langsung menebak teknik yang digunakan dan bagaimana cara memecahkannya.

Poin-poin terpenting yang diperhatikan antara lain letak spasi, dan posisi simbol lain selain huruf. Kebanyakan kriptografi jaman dahulu hanya memproses huruf sehingga letak spasi dan tanda baca lainnya dapat memberikan petunjuk tentang panjangnya kata, dan awal atau akhir rangkaian kata. Tanda baca seperti tanda tanya (?) juga dapat memberikan sedikit petunjuk dengan kata yang membuka kalimat karena biasanya kalimat tanya selalu diawali dengan kata-kata tertentu.

4. Metode penyamaran

Pesan yang disandikan dengan menggunaka kriptografi jaman dahulu pasti telah mengetahui kelemahan metodenya, sehingga biasanya pesan dikirim dengan sedikit perubahan. Salah satu caranya ialah dengan menghilangkan spasi dan tanda baca pada pesan. Penghilangan spasi dan tanda baca pada pesan akan menyulitkan seorang kriptanalis untuk menganalisis pesan, sedangkan bagi penerima pesan yang telah mengetahui kuncinya, pesan tanpa spasi, meskipun lebih sulit untuk dibaca tetapi masih dapat diketahui maksudnya dengan mudah. Tanda baca dan angka juga dapat dituliskan langsung pada pesan, misalnya tanda baca titik (.) ditulis pada pesan sebagai 'TITIK'. Sebetulnya metoda ini dipakai karena pada jaman dahulu, pengiriman pesan umumnya masih memakai jasa kawat yang hanya dapat mengirimkan huruf saja.

5. *Unbreakable Cipher*

Pembuatan *cipher* yang tidak dapat dipecahkan masih menarik. Pada teknik enkripsi saat ini, algoritma untuk memecahkan pesan sebenarnya sudah ada, tetapi karena menggunakan perpangkatan bilangan yang sangat besar, membuat dibutuhkan waktu perhitungan yang sangat lama. Satu-satunya *cipher* yang belum diketahui algoritma pemecahannya adalah *One-time Pad*. *Cipher* ini berbasis *Vigenere cipher* hanya saja kata kunci yang dipakai panjangnya sama dengan panjang pesan yang

akan disandikan.

Prinsip utama *unbreakable cipher* adalah penyusunan metode dekripsi yang memberikan peluang yang sama bagi setiap simbol untuk menggantikan simbol lainnya. Hal ini yang menjadi pertimbangan dalam mengembangkan *Sonics Cipher* karena dengan mempelajari *One-time Pad* dan *Vigenere cipher*, yang menjadi perhatian adalah bagaimana cara membangkitkan suatu kata kunci yang mempunyai panjang yang sama dengan panjang pesan tanpa menggunakan perulangan.

6. Pengembangan *Sonics Cipher*

6.1. Persiapan

Sonics cipher menggunakan sebuah set simbol yang telah ditentukan terlebih dahulu. Pada pengembangan, set simbol yang digunakan hanyalah alfabet tanpa membedakan huruf besar dan kecil ('A'..'Z'), tanda spasi (' '), dan angka ('0'..'9'). Set ini terdiri dari 37 simbol (26 huruf + 10 angka + 1 spasi). Alasannya adalah sebuah pesan sederhana sudah dapat dienkripsikan dengan baik hanya dengan set simbol ini.

Semakin banyak simbol yang digunakan dalam suatu set sebenarnya mempersulit proses kriptanalisis, tetapi juga mempersulit proses enkripsi dan dekripsi pesan sehingga tidak terlalu diperlukan. Selain itu juga, jika pesan dikirimkan dengan menggunakan metode lama seperti kawat atau kode morse, maka akan sulit untuk mengirimkan tanda baca dan perbedaan antara huruf besar dan huruf kecil.

Set simbol ini kemudian diatur mejadi sebuah deret. Deret ini berfungsi sama seperti *Caesar wheel* dimana urutan setiap simbol di dalam deret pasti sama, kecuali deret ini dapat digeser seperti *Caesar wheel* dan berfungsi sebagai penunjuk simbol pengganti. Jika diketahui sebuah deret:

$$\mathbf{D_A}=\{1,2,3,4,5\}$$

Penggeseran deret tersebut sejauh 2 ke arah kiri akan menghasilkan sebuah deret baru:

$$\mathbf{D_B}=\{3,4,5,1,2\}$$

Perhatikan bahwa pergeseran deret ini berfungsi sama persis dengan pergeseran dari *Caesar wheel.*

Setiap simbol mempunyai nilai indeks yang merupakan nilai acuan utama. Pada pengembangan *Sonics Cipher*, sebuah standar yang disebut *Sonics Cipher 1 (SC1)*, mendefenisikan sebuah deret yang hanya terdiri dari huruf, spasi, dan angka yaitu:

$\mathbf{D_{SC1}}=$
{'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z',' ','0','1','2','3','4','5','6','7','8','9'}

Maka huruf 'A' dalam deret *SC1* memiliki indeks 0, huruf 'B' memiliki indeks 1, dst.

Perbedaan antara *Caesar wheel* dan *Sonics Cipher* adalah *Sonics Cipher* menggunakan tiga deret. Deret pertama dan kedua berfungsi sama seperti pada *Caesar wheel* yaitu sebagai penunjuk simbol subtitusi dan simbol yang akan disubtitusi, sedangkan deret ketiga digunakan sebagai pembangkit kata kunci.

6.2. Peng-enkripsian

Algoritma enkripsi *Sonics Cipher* sebenarnya

cukup sederhana dengan ide dasarnya adalah penggabungan dari *Caesar cipher, Vigenere cipher,* dan *One-time Pad.*

Karena pengembangan *Sonics Cipher* dan perangkat lunak *Sonics D/Encryptor* dikembangkan dalam lingkungan *Microsoft Visual Studio 2005*, maka pengoperasian variabel mengikuti tata bahasa *Visual Basic*. Set deret yang digunakan adalah set *SC1*.

Pada makalah ini kata kunci disebut dengan ***Password***, pesan asal dengan ***Plaintext***, dan pesan hasil penyandian dengan ***Ciphertext***. Ketiga variabel diatas bertipe *string* sehingga ***Password****(x)* berarti huruf ke-x dari kata kunci awal. Karena *Sonics Cipher* mempunyai 3 buah deret, maka deret tersebut disebut dengan $\mathbf{D_1}$, $\mathbf{D_2}$, $\mathbf{D_3}$. $\mathbf{D_1}$(x) menyatakan anggota ke-x dari deret pertama. Semua indeks dimulai dari 0.

Langkah-langkah pengenkripsian *Sonics Cipher* adalah:

- Membuat suatu variabel **N**, dimana **N** adalah panjang kata kunci awal.

  $N = Password.Length$
- Membuat variabel M1, dimana M1 adalah nilai indeks karakter terakhir ***Password*** dalam deret set yang digunakan dibagi sisa dengan **N** dan ditambah 1. Penambahan 1 diperlukan agar tidak terjadi pembagian dengan 0 pada operasi berikutnya.

  $M1 = (Index(Password(Last)) \bmod N) + 1$
- Membuat variabel M2 dimana M2 adalah nilai indeks karakter pertama ***Password*** dibagi sisa dengan **M1** dan ditambah dengan 1.

  $M2 = (Index(Password(0)) \bmod M1) + 1$
- Kemudian lakukan *loop* ini pada ***Plaintext***, dengan **i** sebagai indeks penanda dan dimulai dari **i=0**:

  - Geser $\mathbf{D_1}$ ke arah kanan sejauh **M1.**
  - Geser $\mathbf{D_3}$ ke arah kiri sejauh **M2.**
  - Cari simbol yang akan disubtitusi di $\mathbf{D_2}$ dan subtitusi simbol tersebut dengan simbol yang ada di $\mathbf{D_1}$, **s**ehingga:

    $Ciphertext(i) = D_1(Index(Plaintext(i)))$
  - Tambahkan kata kunci dengan simbol yang ada di $\mathbf{D_3}$, sehingga:

    $Password(Last+i) = D_3(Index(Plaintext(i)))$
  - Ganti nilai **M1** dengan nilai indeks ***Password(i)***.

    $M1 = Index(Password(i))$
  - Ganti nilai **M2** dengan nilai indeks dari ***Ciphertext(i)***

    $M2 = Index(Ciphertext(i))$
  - Ulangi langkah-langkah ini untuk simbol selanjutnya. Perhatikan bahwa $\mathbf{D_2}$ tidak dirubah sama sekali karena $\mathbf{D_2}$ berfungsi sebagai acuan untuk melihat simbol ***Plaintext***.

Untuk melakukan proses dekripsi, langkah-langkah yang digunakan sama dengan proses enkripsi, kecuali pada saat melakukan penggeseran deret. Untuk dekripsi, geser $\mathbf{D_1}$ ke arah **kiri** sejauh **M1** dan geser $\mathbf{D_3}$ ke arah kiri sejauh **M1+M2.** Jika pesan yang disandikan dianggap sebagai ***Plaintext***, maka ***Ciphertext*** akan menghasilkan pesan awal.

6.3. Hasil penyandian

Untuk melakukan pengetesan, kalimat

```
'UJI COBA SONICS CIPHER'
```

akan disandikan dengan menggunakan kata kunci

```
'TEST'
```

- Variabel **N** bernilai 4.
- Nilai indeks dari 'T' adalah 19, maka nilai **M1=(19 mod 4) +1 = 4**
- Nilai **M2=(19 mod 4) +1 = 4**
- **D_1** digeser ke arah kanan sebanyak 4.
- **D_2** digeser ke arah kiri sebanyak 4
- Nilai pada **D_1** yang bersesuaian dengan nilai **D_2** untuk huruf 'U' adalah 'Q' sehingga U diganti dengan Q.
- Nilai pada **D_3** yang bersesuaian dengan nilai **D_2** untuk huruf 'U' adalah 'Y' sehingga ***Password*** ditambah dengan 'Y'.
- Nilai indeks dari 'T' adalah 19 sehingga nilai **M1** diganti dengan 19.
- Nilai indeks dari 'Q' adalah 16 sehingga **M2** diganti dengan 16.
- **D_1** digeser ke arah kanan sebanyak 19
- **D_2** digeser ke arah kiri sebanyak 16
- Nilai pada **D_1** yang bersesuaian dengan nilai **D_2** untuk huruf 'J' adalah 'X' sehingga J diganti dengan X.
- Nilai pada **D_3** yang bersesuaian dengan nilai **D_2** untuk huruf 'J' adalah '2' sehingga ***Password*** ditambah dengan '2'.
- Nilai indeks dari 'E' adalah 4 sehingga nilai **M1** diganti dengan 4.
- Nilai indeks dari 'X' adalah 23 sehingga **M2** diganti dengan 23.
- Pada karakter ke-4 yaitu spasi (' '), proses enkripsi tetap dilakukan seperti biasa, tidak seperti teknik enkripsi lain yang membiarkan spasi. Pada *SC1*, spasi mempunyai indeks 26, sehingga bila dilihat dari posisi **D_1** saat ini, nilai yang berada pada **D_1(26)** adalah 'S', sehingga spasi pada ***Plaintext(4)*** diganti dengan huruf 'S" di ***Ciphertext(4).*** Proses penghitungan **M1** dan **M2** dilanjutkan seperti biasa.
- Pada saat mengenkripsi karakter ke-5, terlihat bahwa ***password*** yang digunakan sudah berbeda. Pembangkitan ***password*** secara otomatis dan berdasarkan pada kata kunci yang asli dan *ciphertext* yang sudah ada sebelumnya menjadi salah satu keunggulan *Sonics Cipher* dalam keamanannya. Teknik ini yang menyebabkan pergantian sekecil apapun pada ***ciphertext*** akan menghasilkan ***plaintext*** yang berbeda, yang akan dibahas pada saat pengujian hasil enkripsi.
- Proses ini diulang hingga ***Plaintext*** mencapai huruf 'R' terakhir.

Hasil dari penyandian dengan *Sonics Cipher* adalah:

Plaintext:
`UJI COBA SONICS CIPHER`
Password:
`TEST`
Ciphertext:
`QXSSMA5R3PR8S4HBVEN7IV`
Password yang dibangkitkan oleh algoritma:
`TESTY2ONH4SMSDO3X8IXA0`
Daftar pergerakan D_1 (M1):
`4,19,4,18,19,24,29,14,13,7,31,18, 12,18,3,14,30,23,35,8,23,0`
Daftar pergerakan D_3 (M2):
`4,16,23,18,18,12,0,32,17,30,15, 17,35,18,31,7,1,21,4,13,34,8`

Hasil lain dari penyandian terhadap kata-kata *Cid Highwind* dari *Final Fantasy VII* tentang *Science and Magic*. Beberapa tanda baca dirubah secara otomatis menjadi spasi karena *SC1* tidak mendukung adanya tanda baca.

Plaintext:

I DON T GIVE A RAT S ASS WHETHER IT S SCIENCE OR MAGICAL POWER NO I GUESS IF I HAD TO CHOOSE I D RATHER PUT MY MONEY ON THE POWER OF SCIENCE HUMANS WHO USED TO ONLY ROAM AROUND ON THE GROUND ARE ABLE TO FLY NOW AND FINALLY WE RE ABOUT TO GO INTO OUTER SPACE SCIENCE IS A POWER CREATED AND DEVELOPED BY HUMANS AND SCIENCE JUST MIGHT BE WHAT SAVES THIS PLANET I WAS ABLE TO EARN MY LIVING THANKS TO SCIENCE SO TO ME THERE S NOTHING GREATER

Password:

CID

Ciphertext:

HX2A0GCJ00CM8SNZ52HDHABWD1Y4
574P5MPAEXPWLR6EOHJTRPVFNUT91TZI
72ZBHSZCNELS8IIYXLS722JKDNIJ932L
8PRF5858HG07COMCNVF8RA2BC29BW87E
U
IP3OODW5IGFN50M97GRJ6TG2RJ524CUY
AM4JSRWAJBD12I7U8 E8
SR35C8Y5K2CM75AWIOAB03RW6JQZJJOJ
YYJ
H5GE50CEKHWA8TGJS2AGCFYZ0AB6BO9P
EWDU3LFH2LDKWNBJA6GPFQ0CX0RNZKGD
XBR5EKJR3L2 6BB
D4BS82CGKVGK64ZG7WKR71EFXNI4174E
QHZBM2UJC57MG7A6YNT3M9K9PLKLOOCS
1R874
5GA1ONZV8FLUUDZ18D3VZ6PNVL4DIZWC
CV0LVP54YFR6IL31AZIBFD4YAHYIIPSV
5Q5D3RR7XIQ0G5CNX6DMI0

Password yang dibangkitkan oleh algoritma:

CIDJ77A9CBK914QNW26ESRQ0I
0UT5QZC4EHEK29HDK4M5B9Z0ASX1M87I
C9YOEL5B3FH
HO70ID86MCX3XWAWRMPNBW9F841L
ZLUE0I71PZ9CTUDK0035XPA4SSQH238N
UGSI4W8OOEPY1HVFC TUE9KKXRTXLLRD
4UEK JW6ZZCG1994 XLX2IOC2Y6I2S
4S9KMHGRTTFHWN72XITU3ZQZZI3WFD8Y
CM5TTNOFFKY8FU4RM50DQZO5PZFU10A0
KLOOJ6L7G8P0BPUM9EE2Q3D8A4OHLPBZ
2SLZN5HVA 797POT
ZIAWEAGZOYK5A1GYA99OLEHASKS3D6XS
QJH4HQTAUS3FYPH0ZN48AP0Q6J6G157D
W4E5B3CI2FKG
2FEO45OETZ6D3GY2YO14C4RF430137R9
2OCC9QMBAXW4CDGVSABZ9Q7TE1ENXJPE
NS9P

Daftar pergerakan D_1 (M1):

1,2,8,3,9,34,34,0,36,2,1,10,36,
28,31,16,13,22,29,33,4,18,17,16,
27,8,26,27,20,19,32,16,25,2,31,
4,7,4,10,29,36,7,3,10,31,12,32,
1,36,25,27,0,18,23,28,12,35,34,
8,2,36,24,14,4,11,32,1,30,5,7,
26,7,14,34,27,8,3,35,33,12,2,23,
30,23,22,0,22,17,12,15,13,1,22,
36,5,35,31,28,11,26,25,11,20,4,
27,8,34,28,15,25,36,2,19,20,3,
10,27,27,30,32,23,15,0,31,18,18,
16,7,29,30,35,13,20,6,18,8,31,
22,35,14,14,4,15,24,28,7,21,5,2,
26,19,20,4,36,10,10,23,17,19,23,
11,11,17,3,26,31,20,4,10,26,9,
22,33,25,25,2,6,28,36,36,31,26,
23,11,23,29,8,14,2,29,24,33,8,
29,18,26,31,18,36,10,12,7,6,17,
19,19,5,7,22,13,34,29,23,8,19,
20,30,25,16,25,25,8,30,22,5,3,
35,24,2,12,32,19,19,13,14,5,5,
10,24,35,5,20,31,17,12,32,27,3,

16,25,14,32,15,25,5,20,28,27,0,
27,10,11,14,14,9,33,11,34,6,35,
15,27,1,15,20,12,36,4,4,29,16,
30,3,35,0,31,14,7,11,15,1,25,29,
18,11,25,13,32,7,21,0,26,34,36,
34,15,14,19,26,25,8,0,22,4,0,6,
25,14,24,10,32,0,28,6,24,0,36,
36,14,11,4,7,0,18,10,18,30,3,33,
23,18,16,9,7,31,7,16,19,0,20,18,
30,5,24,15,7,27,25,13,31,35,0,
15,27,16,33,9,33,6,28,32,34,3,
22,31,4,32,1,30,2,8,29,5,10,6,
26,29,5,4,14,31,32,14,4,19,25,
33,3,30,6,24,29,24,14,28,31,2,
31,17,5,31,30,27,28,30,34,17,36,
29,14,2,2,36,16,12,1,0,23,22,31,
2,3,6,21,18,0,1,25,36,16,34,19,
4,28,4,13,23,9,15,4,13,18,36

Daftar pergerakan D_3 (M2):

1,7,23,29,0,27,6,2,9,27,27,2,12,
35,18,13,25,32,29,7,3,7,0,1,22,
3,28,24,31,26,32,34,31,15,32,12,
15,0,4,23,15,22,11,17,33,4,14,7,
9,19,17,15,21,5,13,20,19,36,28,
19,25,8,34,29,25,1,7,18,25,2,13,
4,11,18,35,8,8,24,23,11,18,34,
29,29,9,10,3,13,8,9,36,30,29,11,
35,15,17,5,32,35,32,35,7,6,27,
34,2,14,12,2,13,21,5,35,17,0,29,
1,2,29,36,1,22,35,34,4,20,26,8,
15,30,14,14,3,22,32,8,6,5,13,32,
27,12,36,34,6,17,9,33,19,6,29,
17,9,32,29,31,2,20,24,0,12,31,9,
18,17,22,0,9,1,3,28,29,8,34,20,
35,26,4,35,26,18,17,30,32,2,35,
24,32,10,29,2,12,34,32,0,22,8,
14,0,1,27,30,17,22,33,9,16,25,9,
9,14,9,24,24,9,26,7,32,6,4,32,
27,2,4,10,7,22,0,35,19,6,9,18,
29,0,6,2,5,24,25,27,0,1,33,1,14,
36,15,4,22,3,20,30,11,5,7,29,11,
3,10,22,13,1,9,0,33,6,15,5,16,
27,2,23,27,17,13,25,10,6,3,23,1,
17,32,4,10,9,17,30,11,29,26,33,
1,1,26,3,31,1,18,35,29,2,6,10,
21,6,10,33,31,25,6,34,22,10,17,
34,28,4,5,23,13,8,31,28,34,31,4,
16,7,25,1,12,29,20,9,2,32,34,12,
6,34,0,33,24,13,19,30,12,36,10,
36,15,11,10,11,14,14,2,18,28,17,
35,34,31,26,32,6,0,28,14,13,25,
21,35,5,11,20,20,3,25,28,35,3,
30,21,25,33,15,13,21,11,31,3,8,
25,22,2,2,21,27,11,21,15,32,31,
24,5,17,33,8,11,30,28,0,25,8,1,
5,3,31,24,0,7,24,8,8,15,18,21,
32,16,32,3,30,17,17,34,23,8,16,
27,6,32,2,13,23,33,3,12,8,27

7. Pengujian *Sonics Cipher*

Semua pengujian yang dilakukan terhadap algoritma *Sonics Cipher* dilakukan dengan menggunakan standar *Sonics Cipher 1 / SC1*.

7.1. Pengujian terhadap frekuensi kemunculan

Pengujian pertama yang dilakukan adalah menguji pesan yang telah dienkripsi dengan teknik menganalisis frekuensi kemunculan. Untuk menguji digunakan beberapa kalimat bahasa Inggris yang akan disandikan. Pemilihan kalimat dalam bahasa Inggris digunakan dengan alasan kalimat dalam bahasa Inggris pasti akan

banyak mengandung huruf 'E' sehingga sangat mudah untuk melakukan analisis frekuensi.

Plaintext:

SHE LOVES TO HAVE GREEN TEA IN GREECE

Password:

TEA

Ciphertext:

RYRCFUS0GIFDOR49ZVC90VXQD29AWFI0VBN8X

Dari kalimat diatas, *plaintext* mengandung 9 huruf 'E'. Pada *ciphertext* simbol yang paling sering muncul adalah 'R', '0', '9', dan 'V' masing-masing sebanyak 3 kali. Selain itu masih terdapat simbol-simbol lain yang menggantikan huruf 'E' pada *plaintext.*

Plaintext:

IT WAS NEARING MIDNIGHT AND THE PRIME MINISTER WAS SITTING ALONE IN HIS OFFICE READING A LONG MEMO THAT WAS SLIPPING THROUGH HIS BRAIN WITHOUT LEAVING THE SLIGHTEST TRACE OF MEANING BEHIND HE WAS WAITING FOR A CALL FROM THE PRESIDENT OF A FAR DISTANT COUNTRY AND BETWEEN WONDERING WHEN THE WRETCHED MAN WOULD TELEPHONE AND TRYING TO SUPPRESS UNPLEASANT MEMORIES OF WHAT HAD BEEN A VERY LONG TIRING AND DIFFICULT WEEK THERE WAS NOT MUCH SPACE IN HIS HEAD FOR ANYTHING ELSE THE MORE HE ATTEMPTED TO FOCUS ON THE PRINT ON THE PAGE BEFORE HIM THE MORE CLEARLY THE PRIME MINISTER COULD SEE THE GLOATING FACE OF ONE OF HIS POLITICAL OPPONENTS THIS PARTICULAR OPPONENT HAD APPEARED ON THE NEWS THAT VERY DAY NOT ONLY TO ENUMERATE ALL THE TERRIBLE THINGS THAT HAD HAPPENED IN THE LAST WEEK AS THOUGH ANYONE NEEDED REMINDING BUT ALSO TO EXPLAIN WHY EACH AND EVERY ONE OF THEM WAS THE GOVERNMENT S FAULT

Password:

HARRY

Ciphertext:

DHO31WTSH1ZYHJ2M7C3G3FUBL854UOF7FINXKAFN4WNAW7YMYZ4XH2NSG6AO9L6LEU7R5G32UE9BP9XO2BYA4C351MYSNXG8QLKSMN7ZT291JYW60HQLM4M50WD4VZU1 YNE4Z1NWX M6XWDSD55U W UCPUK13NR1ADKY3S1L6SQZ8JAV
NIU12UY9OYVJ86856KME7AAQKLNT7HJOS
T0LKW5R9Z46MZXBLHNRU1GCJMS5M4FJ0XY96V6Q
X1W1V0Z622Z1FGA79OGXWMHCCSA747TJ2J1DJ8BLP9RW9M90CA7F2O76GTG688YOH51IW0X7SJ3X1HBJES
QLMOKJ5ZE3WBDBTZ84JJFACY7Y1FT6CXU0 6X9QUSGOA9W1E50GR2
A9XMOIC0E26ZUWATADIGYP2
T0V18PUDJDQ TK2I8BHCL
KUOKHMI2ZIXU14XON45X QOLL
R8TB90WMUJB3GDU5XPQ0O
C3RSPVWRXBI0YNYL6XF4ATMHRY
0GLJ54AFIQHMJSL8GM
WECHIKZO8D46TYOOZN
HRVHG8L9H17D1DPMMGIPHMNOR3OOIYV2QP3AIR0DJQC
NMDZ9DRDND22XSFGYATRG7ZECXZN11OY

```
1GM7HIBKM
L0Y70EWCWUHLV93FER6ZQCBW2Z1X
L4PHJAPBELIAZCNQFK19CWWDL4GNZ354
N77 UY8J1B2CQRJRCL2GRD35E
X9BPGWC21X04JPC6SVHTPDM0TUK
WAUS9EOCXA9SS4SY7AI5
R3FOOIJQ9J6FB7SEKIKZVYNILX8Z8UBD
LOA86T5QXEAA5136FNIVX IH5YK4
7F8PNXD4PM2DKD387SH1T8NV8XOK1ZSR
CSS9K9JW0S7
745VWDNCH4RYQ19XBJHOTFVGB009
```

Plaintext adalah paragraf pertama dari bab pertama buku *Harry Potter and the Half Blood Prince* karangan J.K. Rowling. Teks tersebut disandikan dengan kata kunci 'HARRY'. *Ciphertext* menunjukkan banyak sekali terdapat angka-angka meskipun pada *plaintext* tidak terdapat angka sama sekali. Pada *plaintext* terdapat banyak spasi ganda. Hal ini disebabkan karena standar *SC1* mengganti semua simbol yang tidak ada dalam deret set dengan spasi.

Posisi spasi pada *ciphertext* juga penting karena spasi memiliki bobot yang sama dengan dengan simbol-simbol lainnya, seperti pada potongan *ciphertext* ini `D55U  W UC` dimana terdapat 2 spasi setelah huruf U. Dari *ciphertext* sangat sulit menentukan simbol subtitusi yang digunakan dengan teknik analisis frekuensi karena kemunculan banyak simbol baru selain huruf yang digunakan membuat persebaran frekuensi kemunculan yang semakin merata.

7.2. Pengujian terhadap penggantian pesan

Pengujian kedua dilakukan dengan mencoba menginterupsi pesan tersandi dengan cara mengganti sebagian pesan tersebut.

Plaintext:
`SEARCHING FOR FRIENDS`
Password:
`CELES`
Ciphertext:
`O806OBSCRP1P43ELDA29W`

Jika beberapa simbol di *ciphertext* diganti dengan simbol lain, (simbol yang diganti ditebalkan)

Ciphertext:
`O806OBSCRP1P`**`5`**`3ELDA29`**`X`**
Password:
`CELES`
Plaintext:
`SEARCHING FOS FRIEOFW`

Terlihat dengan jelas bahwa pesan yang didapatkan rusak. Jika diperhatikan dengan lebih baik, pesan asli mulai rusak 5 karakter setelah karakter pertama diganti pada *ciphertext*. Hal ini dapat dijelaskan jika melihat ***password*** yang dibangkitkan oleh algoritma:

Password encrypt:
`CELESVVPWDWYKFF99RU2I`
Password decrypt:
`CELESVVPWDWYKFF99`**`SV3J`**

Huruf 'K' yang diberi garis bawah adalah posisi dimana pesan pertama kali diganti. Tepat 5 karakter sesudahnya, ***password*** yang dibangkitkan berbeda. Hal ini disebabkan karena ***password*** awal yaitu 'CELES' memberikan jeda sebanyak **N** karakter sebelum simbol yang

dibangkitkan dari $\mathbf{D_3}$ dipakai.

Dari pengujian terlihat bahwa penggantian satu atau beberapa karakter pada *ciphertext* akan mengakibatkan pesan asli menjadi rusak, tetapi yang berbeda adalah posisi tempat simbol diganti dan posisi setelah **N** karakter tempat simbol diganti untuk pertama kalinya sampai dengan akhir pesan. **N** adalah panjang kata kunci. Sifat ini dapat dimanfaatkan untuk pembangkitan kembali pesan awal selama simbol yang diganti tidak banyak. Seperti pada contoh dimana kata 'FOR' telah rusak dan menjadi 'FOS'. Pada bahasa Inggris tidak terdapat kata 'FOS' dan karena itu dapat menebak jika kata aslinya adalah 'FOR' dan jika pesan yang telah dimodifikasi didekripsi ulang, pesan asli akan muncul.

Ciphertext:
O806OBSCRP1P43ELDA29**X**
Password:
CELES
Plaintext:
SEARCHING FOR FRIENDT

Pesan yang sudah dibenarkan akan menghasilkan kalimat 'SEARCHING FOR FRIENDT'. Disini terlihat pesan sudah kembali ke bentuk aslinya kecuali kata 'FRIENDT' yang menandakan bahwa masih ada bagian *ciphertext* yang sudah diganti.

7.3. Pengujian terhadap penghilangan pesan

Pengujian ketiga dilakukan dengan mencoba menginterupsi pesan tersandi dengan cara menghilangkan sebagian pesan tersebut.

Plaintext:
SEARCHING FOR FRIENDS
Password:
CELES
Ciphertext:
O806OB**S**CRP1P43ELDA29W

Jika huruf 'S' pada *ciphertext* dihilangkan maka pesan hasil enkripsi akan menjadi

Ciphertext:
O806OBCRP1P43ELDA29W
Password:
CELES
Plaintext:
SEARCH21EC23AU2926Q1

Terlihat dengan jelas bahwa pesan hasil dekripsi langsung rusak ketika menemui bagian yang dihilangkan. Dengan cara ini penginterupsi dapat membuat pesan yang ingin disampaikan tidak dapat didekripsi oleh penerima pesan.

Dengan mengetahui posisi pesan yang diganti (dan untungnya sangat mudah karena posisi tersebut terlihat dengan jelas pada saat dekripsi) maka sebuah karakter tambahan dapat dimasukkan sebagai pengganti karakter yang hilang.

Ciphertext:
O806OB CRP1P43ELDA29W
Password:
CELES
Plaintext:
SEARCHQNG FOZF2MLPDR0

Pada pengujian, simbol pada *ciphertext* yang dihilangkan diganti dengan spasi. Hasil dekripsi ulang menunjukkan bahwa sebagian pesan awal sudah mulai terlihat kembali dan dari tahap ini, rekonstruksi ulang pesan dapat dilanjutkan seperti pada saat terjadi penggantian karakter pada *ciphertext*.

7.4. Pengujian terhadap kemungkinan penambahan pesan

Pengujian keempat dilakukan dengan mencoba menginterupsi pesan tersandi dengan cara menambahkan sebagian pesan tersebut.

Plaintext:
I PREFER THE TERM TREASURE HUNTING
Password:
LOCKE
Ciphertext:
DKWW96WEU65DEPFKNG0TAJ8S7PZ8NIB3B1

Disisipkan karakter '0' pada *ciphertext.*

Ciphertext:
DKWW96WEU65DEP**0**FKNG0TAJ8S7PZ8NIB3B1
Password:
LOCKE
Plaintext:
I PREFER THE T MJ68ZI3IAB49AQBCMX2V

Terlihat bahwa pesan yang dihasilkan mirip dengan tanda-tanda bahwa pesan telah dihilangkan. Jika metode yang sama digunakan untuk mencoba merekonstruksi pesan maka yang terjadi adalah

Ciphertext:
DKWW96WEU65DEP**X0**FKNG0TAJ8S7PZ8NIB3B1
Password:
LOCKE
Plaintext:
I PREFER THE TW7E3FEM L814DX17LHIXN

Plaintext yang telah diberi penambahan karakter 'X' sebagai pengganti karakter yang "hilang" tidak memberikan hasil yang sama seperti pada pengujian ketiga. Karena itu cara kedua dapat dilakukan dengan menghilangkan karakter pada posisi yang membuat pesan rusak, yaitu karakter '0' dan 'X' yang sebelumnya ditambahkan.

Ciphertext:
DKWW96WEU65DEPFKNG0TAJ8S7PZ8NIB3B1
Password:
LOCKE
Plaintext:
I PREFER THE TERM TREASURE HUNTING

7.5. Pengujian terhadap perulangan

Pengujian kelima dilakukan karena kelemahan dari *Vigenere Cipher* yang turut menjadi dasar dari *Sonics Cipher* adalah terjadinya perulangan dalam pesan hasil enkripsi untuk kata yang sama dalam selang tertentu, yaitu jika selangnya merupakan kelipatan dari panjang kata kunci.

Plaintext:
KRIPTOGRAFIKRIPTOGRAFIKRIPTOGRAFIKRIPTOGRAFIKRIPTOGRAFIKRIPTOGRA

FIKRIPTOGRAFIKRIPTOGRAFIKRIPTOGR
AFIKRIPTOGRAFIKRIPTOGRAFI

Password:

TEST5

Ciphertext:

H5TI33K5DH78TDZ91SWZDYNY0EWZ9JNF
YZOOBE0VCYK FH8UKP
H9I1FRN8NIHNU27348WSKAJFSDLWGOJL
FA11IU7Q4KXXP1DPVRABD04
W6AF6DRBOWAAMB

Hasil pengujian tidak memperlihatkan munculnya perulangan di dalam *ciphertext* seperti yang terjadi jika menggunakan *Vigenere Cipher.*

7.6. Pengujian terhadap kemungkinan perubahan kunci

Pengujian terakhir dilakukan dengan mencoba beberapa variasi kata kunci yang mendekati kata kunci asli untuk melihat hasil dekripsi pesan.

Plaintext:

I PREFER THE TERM TREASURE HUNTING

Password:

LOCKE

Ciphertext:

DKWW96WEU65DEPFKNG0TAJ8S7PZ8NIB3
B1

Percobaan pertama dengan menghilangkan karakter terakhir dari kata kunci.

Password:

LOCK

Plaintext:

GYNPCI0B67 FKYBSH40X30EOG
83B9MLJ5

Hasil dekripsi tidak menunjukkan sesuatu yang berarti.

Percobaan kedua dengan menggunakan kata kunci yang memiliki satu karakter yang salah.

Password:

GOCKE

Plaintext:

IVKM9A9HL9TL1LSLSC7YZPCR2X2ZWM3B
JU

Password:

LORCE

Plaintext:

I
P5TUT5EMP0088EHHBP6Q2M1N0YL6NLR0

Password:

LOCKR

Plaintext:

GYNPCQO 7 NWSVP9B7UVU9HSRJL25Y4 R6

Dari ketiga hasil pengujian diatas terlihat jelas bahwa tanpa kata kunci yang tepat, hasil *plaintext* yang didapatkan tidak berarti. Yang paling menunjukkan kemiripan dengan teks asli adalah contoh kedua dimana kata kunci yang digunakan salah pada karakter ketiga.

Plaintext:

I PREFER THE TERM TREASURE HUNTING

Password:

LOCKECOLE

Ciphertext:

DKWW96320Q47
C8W6UWJSIB69SSEJL3S66

Password:
LOCKECOSE
Plaintext:
I PREFER6 OL6 LYT66B58X593WKAOL8IN

Dari hasil di atas terlihat bahwa awal pesan sudah menyerupai pesan asli. Dari semua tes yang dilakukan diketahui bahwa pesan semakin mirip dengan aslinya jika huruf awal, huruf akhir, dan panjang kata kunci sama dengan kata kunci yang asli. Jika salah satu dari ketiga syarat diatas tidak terpenuhi maka pesan yang dihasilkan benar-benar berbeda dengan pesan awal.

Jika ketiga syarat diatas terpenuhi maka sebagian pesan asli akan terdekripsi dengan benar di bagian awal sampai pada karakter dimana kata kunci salah dan setelah itu pesan asli tidak terbaca lagi. Untungnya karena masih merupakan awal pesan sehingga biasanya sulit untuk mendapatkan karakter kata kunci yang benar.

8. Pemodifikasian teknik enkripsi

Sonics cipher dibuat dengan teknik pengenkripsian yang dapat dimodifikasi sehingga semakin menyulitkan kriptanalis untuk memecahkan sandi. Cara paling mudah untuk meningkatkan performa *Sonics Cipher* adalah dengan merubah set deret yang menjadi landasan utama untuk subtitusi.

Standar lain selain *SC1* adalah *Sonics Cipher 1Xtended / SC1X*. *SC1X* memiliki deret sepanjang 256 karakter sesuai dengan urutan karakter ASCII. Pemilihan deret ini didasari dengan perkembangan komputer saat ini yang menyimpan data dalam bentuk *bit* dan *byte*. Dalam 1 *byte* terdapat 256 kemungkinan nilai yang setiap kemungkinan disimpan dalam bentuk set deret $\mathbf{D_{SC1X}}$. Dengan demikian *file* dalam format apapun dapat dibaca sebagai *file of bytes* dan dapat dienkripsikan. Kekurangan metoda ini adalah *file* hasil enkripsi tidak dapat dibuka karena *header* file tersebut juga dienkripsikan, sehingga untuk mendapatkan *file* asli, file hasil enkripsi perlu didekripsi secara keseluruhan.

9. Kesimpulan

Kesimpulan dari makalah ini adalah *Sonics Cipher* saat ini merupakan algoritma enkripsi dengan kunci simetris dengan tingkat keamanan dan kecepatan yang tinggi. Kriptografi kunci simetris sendiri berarti bahwa kata kunci yang dipakai untuk mengenkripsi dan mendekripsi pesan sama, sehingga cocok untuk penggunaan personal atau di dalam suatu kelompok yang memiliki kata kunci bersama.

Sonics Cipher melakukan enkripsi dengan menggunakan perpaduan dari *Caesar Cipher, Vigenere Cipher, dan One-time Pad.*

Caesar cipher diadaptasi dengan penggunaan penggeseran set deret yang digunakan sebagai basis dari *Sonics Cipher*. *Vigenere Cipher* diadaptasi dengan penggunaan kata kunci yang merupakan penunjuk besarnya pergeseran deret yang perlu dilakukan pada setiap langkah. *One-time Pad* dijadikan referensi sebagai satu-satunya teknik enkripsi yang belum dapat dipecahkan sampai saat ini. Keunggulan

One-time Pad dicoba untuk diadaptasi dengan adanya pembangkitan susunan ***password*** yang digunakan dalam enkripsi sehingga menyulitkan proses kriptanalisis pesan. Meskipun mungkin ***password*** yang dibangkitkan tidak sebaik *One-time Pad* dalam hal tingkat kemungkinan pergantian simbolnya, tetapi sudah dirasakan cukup karena membuat proses kriptanalisis dari tengah pesan sulit.

Hasil pengujian terhadap *Sonics Cipher* juga menunjukkan bahwa teknik enkripsi ini cukup baik dalam mengetahui adanya usaha perubahan pesan karena perubahan sekecil apapun dapat merusak pesan. Salah satu keunggulan *Sonics Cipher* juga dari adanya kemungkinan untuk merekonstruksi ulang pesan yang telah mengalami perubahan kecil. Dari tes terhadap penyisipan, penghapusan, dan penggantian *ciphertext*, hasil *plaintext* menunjukkan tanda-tanda perusakkan di tempat pertama terjadinya usaha interupsi pesan. Kemungkinan untuk menambah kemampuan dari *Sonics Cipher* juga menjadi nilai tambah dari algoritma ini. Saat ini baru ada dua standar set deret yang digunakan yaitu *Sonics Cipher 1 (SC1)* yang terdiri dari huruf kapital, spasi, dan angka saja dan *Sonics Cipher 1Xtended (SC1X)* yang terdiri dari 256 karakter ASCII.

10. Daftar Pustaka

[RIN] Munir, Rinaldi, *Diktat Kuliah IF5054 Kriptografi*, Institut Teknologi Bandung 2006

[WIK] *www.wikipedia.org*. Diakses selama bulan September-Oktober 2006.