Studi dan Implementasi Komputerisasi Mesin Enigma

Akhmad Ratriono Anggoro 13505003
Program Studi Teknik Informatika ITB, Jalan Ganesha 10 Bandung, Jawa Barat,
email: if15003@students.if.itb.ac.id

Abstract **–** Makalah ini membahas tentang studi terhadap mesin enigma, teknologi enkripsi yang berupa mesin mekanik-elektrik untuk mengubah pesan teks berupa kalimat dan kata menjadi sandi-sandi rahasia dan mengimplementasikannya dalam bahasa pemrograman modern (dalam kasus ini, Visual basic). Mesin ini banyak digunakan sewaktu perang dunia kedua sedang berlangsung terutama oleh pihak *Axis*.
Sejak pertama kali dibuat pada awal tahun 1920an mesin ini telah memiliki begitu banyak varian baik yang digunakan secara komersil maupun yang dipakai oleh pihak militer untuk mengirim pesan demi kepentingan tertentu. Varian milik militer Jerman termasuk versi yang paling banyak didiskusikan. Versi ini terkenal sebab ahli kriptologi dari pihak sekutu berhasil mendekripsikan banyak sandi rahasia yang dihasilkan oleh mesin tersebut.
Alasan utama keberhasilan tersebut adalah berkat jasa dari 3 orang kriptografer Polandia Marian Rejewski, Jerzy Rozycki dan Henryk Zygalski pada tahun 1932. 7 tahun kemudian, hasil rekonstruksi mesin dan metode dekripsi disebarkan ke Inggris dan Prancis dengan kode nama ULTRA. Pengaruh utama dari ULTRA dapat diperdebatkan sampai sekarang namun banyak yang percaya kalau hasil pengiriman ini memperpendek perang eropa sampai 2 tahun.
Secara gamblang, enigma memang memiliki banyak kelemahan namun sebenarnya keberhasilan ini lebih dikatalisasi dari factor-faktor manusia seperti penyitaan mesin dan buku sandi, interogasi tawanan perang dan operator, serta kesalahan prosedural dalam pengiriman sandi.

Kata Kunci*: mesin enigma, rotor, reflector, plugboard, pemutasi kode, bombe, kode visual basic.*

1. PENDAHULUAN

Sebagai sebuah mesin mekanik elektrik, enigma menggunakan rotor dan rangkaian listrik yang secara konstan memanfaatkan kombinasi aliran listrik dan gerakan rotor untuk mengubah sebuah huruf menjadi kode rahasia. Bentuknya sendiri tak ubah layaknya sebuah mesin tik yang biasa kita temukan beberapa tahun yang lalu, tentu saja dengan modifikasi yang sesuai seperti pemakaian rotor dan *plugboard* sebagai komponen utama penyandian.
Enigma sendiri pada dasarnya menggunakan logika paling sederhana dalam penyandian yaitu subtitusi, mengganti sebuah huruf asal menjadi tepat satu huruf yang berbeda. Namun, semua bisa menjadi berbeda apabila subtitusi satu ke satu itu dilakukan oleh 3 (atau lebih) rotor dengan 26 *node* yang masing-masing berputar layaknya odometer. Di sinilah keindahan enigma, tanpa mesin yang sama, pengaturan posisi rotor yang sama, dan tipe subtitusi yang sama, sebuah kode yang dibuat dengan mesin enigma akan sangat sulit untuk dipecahkan.
Untuk membuktikannya kita lihat persamaan yang berhasil dirumuskan oleh Tuan James Gillogly

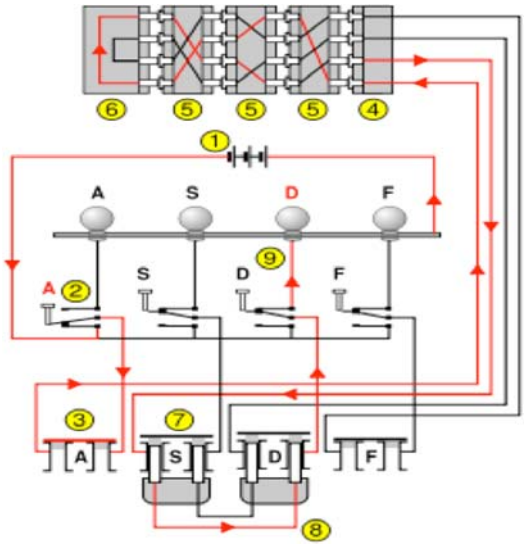
(1)

$$IC = \frac{\sum_{i=A}^{Z} f_i(f_i-1)}{N(N-1)}$$

Jika fi adalah frekuensi dari huruf I dalam sandi dan N adalah total huruf dari sandi maka IC adalah perhitungan kasar dari distribusi huruf, nilai yang besar menunjukkan huruf yang sering dipakai. Sebagai contoh untuk teks random IC yang didapat kira-kira 0.038 sedangkan untuk teks bahasa Jerman nilai yang didapat rata-rata adalah 0,07.

2. CARA KERJA

Mesin Enigma (lengkap dengan reflector dan plugboard) bekerja dengan cara sebagai berikut

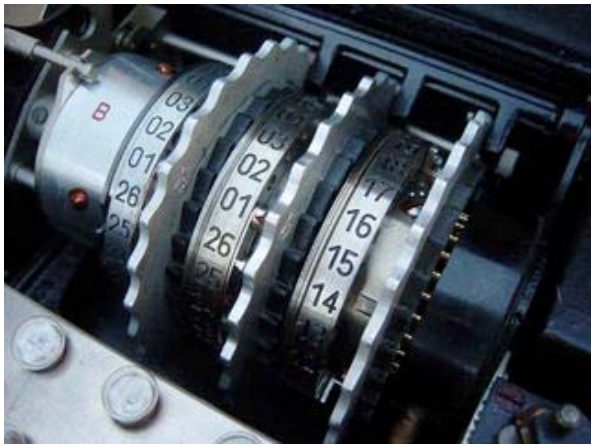


Gambar 1 Bagaimana mesin enigma bekerja

Anggap sebuah mesin enigma dengan 4 huruf yang akan disubtitusi tepat dengan 4 huruf lainnya (dalam hal ini huruf tersebut adalah a, s, d, f). Arus mengalir

dari sumber tenaga batere (1), melewati tombol yang ditekan oleh operator (2), ke plugboard (3), Plugboard mengganti arus awal ke arus huruf yang diinginkan (dalam kasus ini A tetap A) (2) menuju ke rotor awal (4) selanjutnya, arus mengalir ke rotor-rotor berikutnya (5) sampai ke reflector (6). Reflektor mengembalikan arus sesuai arus keluar dari rotor masuk dan menuju plugboard S (7) dan menggantinya ke huruf pengganti di plugboard (S menjadi D) (8) lampu huruf sandi menyala (9). Proses dekripsi pada mesin enigma sama dengan proses enkripsinya

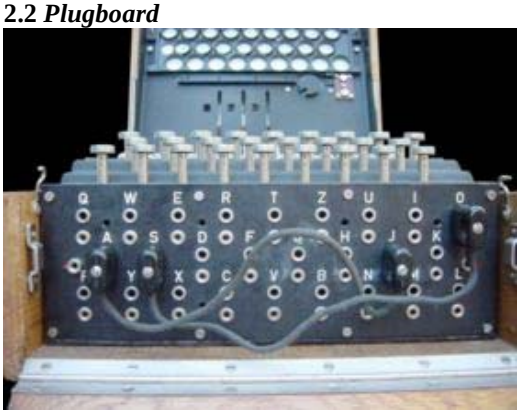
2.1 Rotor



Gambar 2 Rotor

Adalah roda gigi dengan diameter sebesar 10cm biasanya terbuat dari karet keras dengan 26 pin di dalamnya dan 26 huruf alphabet atau angka yang diwakilinya pada sisi luar rotor. Roda ini dapat diputar manual dengan tangan untuk menentukan posisi awal sebelum proses enkripsi dimulai.
Arus yang masuk dalam proses enkripsi akan membuat rotor paling kanan akan berputar sekali dan layaknya odometer, rotor selanjutnya akan berputar ketika rotor awal telah berputar satu putaran penuh.
Satu hal yang perlu menjadi catatan, dalam awal pengembangannya, mesin enigma tidak memiliki reflector seperti yang dijelaskan pada cara kerja sehingga terdapat arus tambahan untuk membalikkan arus masuk yang meyebabkan persentase kesalahan dalam enkripsi menjadi besar. Reflektor sendiri adalah rotor dengan satu pin yang tidak berputar yang membalikkan arus dan mencegah satu huruf disubtitusi dengan huruf yang sama. Setelah ditemukan reflektorlah semua menjadi lebih efektif.

2.2 *Plugboard*



Gambar 3 Plugboard

Berapapun rotor yang digunakan dalam sebuah mesin Enigma tidak akan berpengaruh jika posisi dan tipe rotor sudah diketahui. Di sinilah plugboard berperan. Plugboard pada dasarnya adalah papan untuk mengganti arus dari huruf awal ke huruf yang diinginkan dengan cara meneruskan arus tersebut dengan kabel.
Seperti yang terlihat pada gambar A terhubung dengan J dan S terhubung dengan O. Ini berarti semua A, baik input keyboard maupun hasil keluaran rotor akan menjadi J dan *vice versa*, demikian juga yang berlaku untuk S dan O.

2.3 Deskripsi Matematis

Adapun deskripsi matematis dari sebuah mesin enigma pada dasarnya adalah sebuah permutasi panjang sebagai berikut:
Asumsikan enigma yang dipakai adalah versi dengan tiga rotor, satu reflector, satu *plugboard* maka persamaan yang dihasilkan adalah
(2)

$$E = PRMLUL^{-1}M^{-1}R^{-1}P^{-1}$$

Dengan E sebagai hasil enkripsi dan P, R, M, L, U berturut-turut adalah Plugboard, tiga rotor, dan reflektor.

2.4 Prosedur pengiriman

Jerman memiliki banyak jaringan pengiriman dengan ketetapan berbeda untuk tiap jaringan. Semua ketentuan ditentukan di awal dengan pemberian buku kode standar bagi para operator di masing-masing titik strategis, hal yang nantinya menjadi titik lemah utama dari dekripsi oleh pihak lawan.
Initial state dari tiap-tiap mesin memiliki aspek-aspek sebagai berikut:

- Pilihan rotor dan peletakan posisi rotor.
- Pemilihan posisi huruf awal pada rotor
- Randomisasi posisi huruf pada rotor
- Posisi *Plugboard*
- Tipe reflector

Pada dasarnya jika urutan alphabet pada rotor tidak diketahui enigma memiliki sekitar 10^{114} kemungkinan dan jika susunan tersebut diketahui enigma masih memiliki sekitar 10^{23} kemungkinan

sehingga pengguna Enigma tentu merasa yakin akan keamanan dari Enigma.
Selain kecermatan dalam initial state, operator-operator Jerman juga memiliki konvensi tentang bagaimana sandi ditulis agar tidak terjadi kalimat-kalimat yang ambigu karena setiap rotor memiliki tidak lebih dari 26 huruf standard dan tidak memiliki tanda baca sama sekali. Misalnya menuliskan X untuk titik dan ZZ untuk koma pada setiap pesan.

2.5 Telaah Pragmatis

Setelah mempelajari keseluruhan aspek dari mesin enigma, dapat diambil beberapa garis besar dari untung rugi menggunakan mesin enigma.
Keuntungan

1) Spesifik, memiliki banyak varian dan aturan main dari tiap-tiap mesin enigma, sehingga tidak mungkin memakai pendekatan yang sama dari mesin maupun operator yang berbeda.
2) Tidak dapat dipecahkan dengan *brute force*.
3) Sederhana namun efektif.

Kerugian

1) Bisa terjadi kebocoran manusiawi, pengungkapan oleh operator, atau penyitaan mesin untuk dipelajari.
2) Apabila sebuah mesin telah berhasil dipecahkan sempurna, varian-varian lain ikut terancam.
3) Arsitektur terbatas menyebabkan minimnya modifikasi

3. HASIL DAN PEMBAHASAN

Semua kerugian yang diungkap di bab sebelumnya memiliki solusi, salah satunya adalah komputerisasi, tidak terbatasnya dunia pemrograman bisa menjadi alternatif mengembangkan kembali enkripsi yang sudah terkubur oleh waktu ke dimensi baru.

3.1 Implementasi enigma dalam VB

Pada implementasi yang saya lakukan, program memiliki arsitektur dasar yang sama dengan mesin enigma hanya saja tidak terdapat sebuah reflector dan plugboard sehingga hasil enkripsi dapat dilihat dari rotor kedua (hanya dua rotor yang digunakan di sini).
Walaupun demikian, saya menambahkan sebuah password yang mengacak rotor. Mengacak disini berarti mengubah kedudukan awal dari posisi huruf pada rotor (initial state) layaknya sebuah mesin enigma.

```
Public Const
sDefaultRotor1 As String =
"ABCDEFGHIJKLMNOPQRSTVUWXYZ "

Public Const
sDefaultRotor2 As String =
"IWEHJKTLZVOPFBNMQRYUASDXC “

Function Encrypt(ByVal sInput As String,
sPASSWORD As String, bExtra As Boolean) As
String
   Dim sRotor1 As String
   Dim sRotor2 As String
   sRotor1 = sDefaultRotor1
   sRotor2 = sDefaultRotor2
   sInput = Replace(sInput, Chr(10), "")
   sInput = Replace(sInput, Chr(13), "")
   sInput = Replace(sInput, vbTab, "")

   If bExtra Then sInput = Replace(sInput, " ", "")

'saya akan memakai password untuk mengacak
rotor:
ScrambleRotors sRotor, sRotor2, sPASSWORD

Dim k As Long ' index karakter
Dim c As String ' menyimpan satu huruf.
Dim i As Long ' index karakter dari string yang
dipilih
Dim sResult As String ' hasil.

sResult = ""
   For i = 1 To Len(sInput)
         'Ambil karakter(i)
         c = Mid(sInput, i, 1)
         ' Find character(i) on the first Rotor:
         k = InStr(1, sRotor1, c,
vbBinaryCompare)
         If k > 0 Then
         'Ambil karakter dari index tersebut dari
rotor ke2 dan tambahkan ke result
            sResult = sResult & Mid(sRotor2, k, 1)
         Else
            ' Tidak ketemu, biarkan saja:
            sResult = sResult & c
         End If
         ' Memutar Rotor pertama ke kiri:
         sRotor1 = LeftShift(sRotor1)
         ' Rotate Rotor kedua to the right:
         sRotor2 = RightShift(sRotor2)
   Next i
   Encrypt = cut_lines(sResult, 40)
End Function

Private Function cut_lines(sInput As String,
lBreakAfter As Long) As StringOn Error GoTo
err1
   Dim sResult As String
   sResult = ""
   Dim l As Long
   For l = 1 To Len(sInput) Step lBreakAfter
      sResult = sResult & Mid(sInput, l,
lBreakAfter) & vbNewLine
   Next l
   cut_lines = sResult
   Exit Function
err1:
```

```
   cut_lines = sInput ' unchanged
   Debug.Print "cut_lines:" & Err.Description
End Function

Function Decrypt(ByVal sInput As String, sPASSWORD As String) As String
   Dim sROTOR1 As String
   Dim sROTOR2 As String
   sROTOR1 = sDefaultROTOR1
   sROTOR2 = sDefaultROTOR2


   Dim l As Long
   l = InStr(1, sInput, sDEF_PREFIX, vbTextCompare)
   If l > 0 Then
      sInput = Mid(sInput, l + Len(sDEF_PREFIX))
   End If
   l = InStr(1, sInput, sDEF_SUFFIX, vbTextCompare)
   If l > 0 Then
      sInput = Mid(sInput, 1, l - 1)
   End If

   sInput = Replace(sInput, Chr(10), "")
   sInput = Replace(sInput, Chr(13), "")
   sInput = Replace(sInput, vbTab, "")

   ' Menggunakan password untuk mengacak ulang rotor:

   ScrambleRotors sROTOR1, sROTOR2, sPASSWORD

   Dim k As Long
   Dim i As Long '
   Dim c As String
   Dim sResult As String
   sResult = ""
   For i = 1 To Len(sInput)
         ' Get character(i)
         c = Mid(sInput, i, 1)
         ' mencari karakter dari rotor 2:
         k = InStr(1, sROTOR2, c, vbBinaryCompare)
         If k > 0 Then
            ' mengambil karakter dengan index sama dari rotor 1 menambahkan ke result:
            sResult = sResult & Mid(sROTOR1, k, 1)
         Else
            ' tidak ketemu biarkan saja:
            sResult = sResult & c
         End If

         sROTOR1 = LeftShift(sROTOR1)
         sROTOR2 = RightShift(sROTOR2)
   Next i
   Decrypt = sResult
End Function

' memutar  Rotor (string).
' karakter pertama ke paling belakang
' sisanya ke kiri.

Function LeftShift(s As String) As String
   If Len(s) > 0 Then LeftShift = Mid(s, 2, Len(s) - 1) & Mid(s, 1, 1)
End Function

'memutar ke kanan berlawanan dengan fungsi sebelumnya
Function RightShift(s As String) As String
If Len(s) > 0 Then RightShift = Mid(s, Len(s), 1) & Mid(s, 1, Len(s) - 1)
End Function

‘ Mengacak Rotor
' Bigger password = better scramble!
Sub ScrambleRotors(ByRef sW1 As String, ByRef sW2 As String, sPASSWORD As String)

Dim i As Long
Dim k As Long
For i = 1 To Len(sPASSWORD)
   For k = 1 To Asc(Mid(sPASSWORD, i, 1)) * i
      sW1 = LeftShift(sW1)
      sW2 = RightShift(sW2)
   Next k
Next i
End Sub
```

2.1 Pembahasan

Menggunakan reverse engineering tidak akan banyak berguna dalam memecahkan Enigma tanpa ketentuan yang sesuai, menggunakan *brute force* juga akan memakan banyak waktu sia-sia. Apalagi untuk mesin enigma dengan rotor yang sangat banyak dan belasan konversi dalam *plugboard*. Satu-satunya cara untuk memecahkan kode untuk enigma adalah dengan mendapatkan sedikit ‘bocoran’ tentang tipe, susunan, dan cara kerja dari varian yang dimaksud karena pada dasarnya enigma adalah enkripsi sederhana yang dilapis-lapis sehingga menghasilkan sesuatu yang sangat brilian, setidaknya di masanya dulu.

Saat ini, sandi enigma dengan beberapa varian dapat dipecahkan dengan kemungkinan hampir 80% (Berdasarkan karya tulis James Gillogly) dengan teknik mahir dan sedikit keberuntungan. Namun, kemungkinan-kemungkinan baru tentu tidak tertutup. Bahasa pemrograman komputer, yang tidak dikenal di jaman perang dunia dapat menjadi gerbang baru hadirnya varian-varian baru enigma dengan kualitas yang cerdas, menutupi kelemahan-kelemahan yang ada pada mesin aslinya.

Contoh sederhana, jika pada mesin asli, untuk membuat rotor dengan karakter lebih dari 26 tentu dibutuhkan ukuran rotor yang besar dengan daya

listrik yang besar pula. Sedangkan dengan bahasa pemrograman, bahkan dapat dibuat sebuah rotor dengan ratusan karakter menyertakan semua kode ASCII. Menambahkan revolusi baru seperti plugboard bukan tidak mungkin, saya sempat terpikir untuk membuat sebuah varian dari Plugboard dengan satu huruf yang saling memetakan dengan 4 huruf lainnya yang dilakukan secara sekuensial.
Hal-hal menguntungkan yang bisa didapat dengan migrasi mesin enigma ke komputer antara lain:

1) Kode sumber mesin bisa terjaga, tidak perlu ada konvensi, perubahan kode sumber program dapat menjadi sebuah aturan main bagi pengguna dengan kepentingan yang sama.
2) Sudah tidak terikat keterbatasan mekanik-elektrik, modifikasi besar-besaran, bahkan mungkin bisa digabungkan dengan teknik enkripsi modern.
3) Waktu pengembangan yang semakin cepat, ini berarti akan lebih banyak varian-varian baru bermunculan dalam waktu singkat.

4. KESIMPULAN

Enigma termasuk salah satu mesin penyandian yang bertahan lama di dunia *cipher*. Karena menyimpan sebuah kerumitan terstruktur di balik konsep sederhana, bukanlah sebuah kerugian untuk mencoba menerapkannya dalam komputer.
Komputerisasi sebuah enkripsi kuno bukanlah sesuatu yang sulit, namun lebih banyak yang memilih untuk mencari dan membuat sebuah orisinalitas dengan berbagai alasan terutama keamanan. Enigma sendiri adalah penyebab utama kekalahan Jerman dalam perang dunia kedua sehingga mungkin kurang begitu dipercaya. Kalaupun ada yang menfasilitasi migrasi penyandian kuno ke dalam komputer pasti bukanlah sebuah hal serius seperti layaknya pendanaan pengembangan sebuah enkripsi baru. Akan tetapi enigma merupakan sejarah besar dalam perkembangan matematika diskrit sebelum komputer digunakan secara luas sehingga patut dipelajari dan diingat.

DAFTAR REFERENSI

[1] http://russells.freeshell.org/enigma/how.html
[2] en.wikipedia.org/wiki/Enigma_Machine
[3] Hamer, David H.; Sullivan, Geoff; Weierud, Frode (July 1998). "Enigma Variations: an Extended Family of Machines", *Cryptologia*, 22(3).
[4] James J. Gillogly, "Ciphertext-only Cryptanalysis of Enigma," Cryptologia, 19 (4), 1995, pp. 405–412.