Implementasi *Biometrics* di Dalam Penggunaan Sidik Jari Untuk Meng-*generate* Kunci pada Algoritma Enkripsi *One Time Pad*

Irdham Mikhail Kenjibriel (13508111)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
irdhamkenjibriel@students.itb.ac.id

***Abstract*— *One Time Pad* (OTP) adalah algoritma yang termasuk dalam algoritma enkripsi klasik dan berada dalam kelompok algoritma kriptografi simetri. Algoritma ini sudah terbukti sangat sulit dipecahkan karena mempunyai panjang kunci yang sama dengan panjang pesannya. Namun hal tersebut menciptakan masalah baru karena kita harus dapat menggenerate kunci secara acak dan sama dengan panjang teksnya. Sedangkan *biometrics* adalah penggunaan karestiristik fisik dari anggota atau bagian tubuh manusia yang unik dalam mengidentifikasi manusia secara individual antara satu dengan yang lainnya. Oleh karena itu, pada makalah ini penulis akan mencoba memanfaatkan penerapan *biometrics* dalam hal ini adalah penggunaan sidik jari untuk dapat menggenerate kunci pada algoritma *One time Pad* yang dirasa akan meningkatkan kefektifan dan keefisienan dalam mengamankan pesan yang ingin dienkripsi .**

Pada makalah ini akan dibahas pula mengenai bagaimana penggunaan sidik jari agar dapat mengenkripsi pesan. Sidik jari mempunyai dua komponen yaitu *veins* dan *ridges* yang letaknya atau posisinya pada sidik jari manusia berbeda satu dengan yang lainnya. Sifat unik pada setiap manusia tersebut yang akan dimanfaatkan penulis agar dapat membuat kunci yang random dan unik untuk setiap pembuatannya. Selain dari pada posisi *veins* dan *ridges*-nya penulis juga akan memanfaatkan posisi *minutiae point* yaitu posisi poin pada saat garis dari suatu *ridges* berakhir.

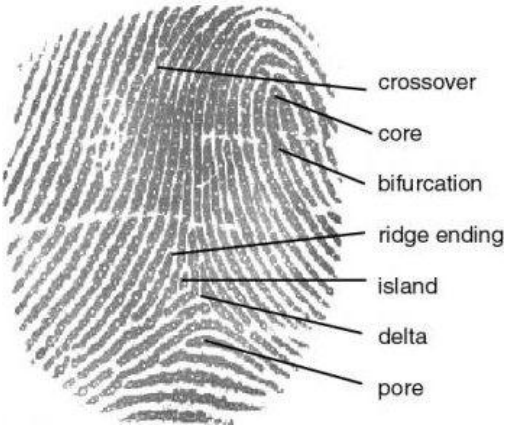
***Index Terms*—** *Biometric, Efektiv, Efisien, Minutiae points, One Time Pad, Ridges, Sidik Jari, dan Veins*

I. PENDAHULUAN

Penggunaan *biometrics* untuk pengamanan dewasa ini sangat banyak digunakan secara umum oleh berbagai kalangan sebagai contoh penggunaan sidik jari pada pengamanan laptop, *face recognition* pada pengamanan laptop, sistem absensi dengan menggunakan sidik jari, dan masih bannyak implementasi biometrics lainnya yang tidak mungkin saya sebutkan satu persatu. Hal tersebut dapat terjadi karena sifat dari salah satu anggota tubuh manusia yang bersifat unik yang dapat dijadikan kunci sehingga berkembangnya *biometrics* bisa menjadi sangat pesat.

Biometrics adalah penggunaan karestiristik fisik dari anggota atau bagian tubuh manusia yang unik dalam mengidentifikasi manusia secara individual antara satu dengan yang lainnya. Retina, DNA, Sidik Jari, dll adalah bagian yang paling sering digunakan untuk keperluan *biometrics.* Karena setiap bagian dari tubuh manusia antara satu dan yang lainnya memiliki komponen tersebut dan berbeda antara individu satu dengan yang lainnya.

Sidik jari adalah salah satu bagian dari tubuh manusia yang paling sering dipakai dalam *biometrics*. Sidik jari setiap manusia memiliki dua komponen atau bagian yaitu *ridges* dan *veins. Ridges* adalah bagian yang berupa daerah dan mempunyai pola seperti pada linggkaran yang tidak sempurna atau obat nyamuk bakar yang garisnya dapat berhenti pada suatu titik dan perhentia dari *ridges* ini yang berupa titik dapat penulis sebut sebagai *minutiae*. Letak antara *minutiae* satu dengan yang lainnya dalam sidik jari berbeda antara setiap individunya. *Veins* adalah bagian yang berupa alur- alur yang membujur yang kadang dapat bercabang pada suatu titik atau menyatu antara *veins-nya*. Di bawah ini adalah gambar yang akan menjelaskan mengenai bagian- bagian dari sidik .



Gambar 1. Komponen- komponen dari sidik jari manusia

Kriptografi berasal dari Bahasa Yunani: *cryptós* artinya rahasia, sedangkan *gráphein* artinya tulisan. Jadi, secara morfologi kriptografi berarti tulisan rahasia.

One Time Pad adalah algoritma yang termasuk dalam algoritma enkripsi klasik dan berada dalam kelompok algoritma kriptografi simetri. Algoritma ini mempunyai kelebihan dari algoritma yang lain dalam hal tingkat keamanannya sebagai algoritma yang paling sulit dipecahkan karena dua hal yaitu:

- *Kunci yang dipilih secara acak (yaitu, setiap kunci harus mempunyai peluang yang sama untuk terpilih).*
- Panjang kunci yang sama dengan panjang plainteks yang akan dienkripsikan.

Namun hal tersebut memiliki beberapa masalah yaitu:

- Karena panjang kunci harus sama dengan panjang pesan, maka *one-time pad* hanya cocok untuk pesan berukuran kecil. Semakin besar ukuran pesan, semakin besar pula ukuran kunci. Pada aplikasi kriptografi untuk mengenkripsikan data tersimpan, timbul masalah lain dalam penyimpanan kunci.
- Karena kunci dibangkitkan secara acak, maka 'tidak mungkin' pengirim dan penerima membangkitkan kunci yang sama secara simultan. Jadi, salah seorang dari mereka harus membangkitkan kunci lalu mengirimkannya ke pihak lain.

Oleh karena itu penulis bermaksud untuk menerapkan prinsip *biometrics*, pemakaian sidik jari, dalam membangkitkan kunci untuk proses enkripsi maupun dekripsi untuk algoritma *One Time Pad* ini. Sehingga diharapkan tingkat kefektifan, keefisenan, dan keamanan dari penggunanan algoritma *One Time Pad* menjadi bertambah dengan dari pada menggenerate kunci dengan menggunakan mesin lainnya.

II. DASAR TEORI

A. One Time Pad Cipher

One-time pads ditemukan pada tahun 1917 oleh Major Joseph Mauborgne. Cipher ini termasuk ke dalam kelompok algoritma kriptografi simetri.

One-time pad (*pad* = kertas bloknot) berisi barisan karakter-karakter kunci yang dibangkitkan secara acak. Aslinya, satu buah *one-time pad* adalah sebuah pita (*tape*) yang berisi barisan karakter-karakter kunci.

Satu *pad* hanya digunakan sekali (*one-time*) saja untuk mengenkripsi pesan, setelah itu *pad* yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain.

- Aturan enkripsi yang digunakan persis sama seperti pada *cipher* Vigenere. Pengirim pesan menggunakan setiap karakter kunci untuk mengenkripsikan satu karakter plainteks.

- Enkripsi dapat digambarkan sebagai penjumlahan modulo jumlah karakter total yang ada dari satu karakter plainteks dengan satu karakter kunci *one-time pads*:

$$c_i = (p_i + k_i) \bmod x$$

yang dalam hal ini,

p_i : karakter plainteks
k_i : karakter kunci
c_i : karakter cipherteks
x : jumlah karakter yang ada dalam hal ini yang dipakai adalah karakter ASCII (256)

Perhatikan bahwa panjang kunci sama dengan panjang plainteks, sehingga tidak ada kebutuhan mengulang penggunaan kunci selama proses enkripsi.

Setelah pengirim mengenkripsikan pesan dengan *one-time pads*, ia menghancurkan *one-time pad* tersebut (makanya disebut satu kali pakai atau *one-time*)

Penerima pesan menggunakan *one-time pads* yang sama untuk mendekripsikan karakter-karakter cipherteks menjadi karakter-karakter plaintek dengan persamaan:

$$p_i = (c_i - k_i) \bmod x$$

yang dalam hal ini,

p_i : karakter plainteks
k_i : karakter kunci
c_i : karakter cipherteks
x : jumlah karakter yang ada dalam hal ini yang dipakai adalah karakter ASCII (256)

B. Pengolahan Sidik Jari dalam Pembentukan Kunci

Pertama penulis akan mengolah kunci dengan mendapatkan kunci dalam bentuk file gambar. File gambar ini disarankan agar dalam kualitas yang baik sehingga komponen- komponen sidik jari yang diperlukan dalam peng-*generate*-an kunci seperti *veins, ridges*, dan *minutiae* dapat terdeteksi dengan jelas. Untuk keperluan tersebut format file gambar yang dipakai adalah berupa *bitmap* file.

Masing- masing baik dari pihak penerima pesan (dalam hal ini penulis sebut sebagai dekriptor) dan pihak yang mengirimkan pesan (dalam hal ini penulis sebut sebagai enkriptor) saling menukarkan sidik jari mereka yang berupa file gambar dengan ukuran pixel dan format (*bitmap* file) yang sama. Setelah itu dari kedua belah pihak memutuskan bagian mana yang digunakan dalam menggenerate kunci dari sidik jari mereka masing input nya berupa dua parameter yang pertama adalah titik awal derah seleksi berupa poin (*x*,*y*) dalam *pixels* dari file gambar dan berapa luasnya lebar dan tinggi file seleksinya daalam ukuran *pixels* juga. Luas dari file gambar yang dipilih menentukan besar pesan yang akan dienkripsi.

Kedua informasi ini harus dijaga dengan ini harus dijaga dengan baik, tidak hanya oleh pihak penerima pesan tetapi juga pihak pengirim pesa. Sehingga walaupun ada orang yang berniat jahat dengan ingin mengetahui pesan yang bukan merupakan haknya dan berusaha mendapatkan gambar sidik jari dan berhasil mereka tetap tidak akan mengetahui bagian mana dari gambar tersebut yang dipakai dalam menggenerate kunci jadi kepemilikan

file gambar tersebutpun menjadi percuma tanpa mengetahui bagian mana yang dipilih dari sidik jari tersebut yang berguna untuk menggenerate kuncinya.

C. Meng-*generate* Kunci dengan Menggunakan Komponen *Veins* dan *Ridges*

Setelah mendapatkan file image dari pihak pengirim pesan (bagi penerima pesan) dan dari pihak penerima pesan (bagi pengirim pesan) maka sekarang baik pihak pengirim maupun penerima akan memiliki dua buah file gambar yang masing- masing adalah gambar sidik jari dari pihak penerima dan dari pihak pengirim.

Hal yang pertama penulis lakukan adalah meng-*generate* kunci dari file gambar sidik jari pihak pengirim. Selanjutnya penulis mengecek satu persatu pixel dari gambar yang telah dipilih atau seleksi dari pixel yang berada di pojok kiri atas sampai pada pixel yang berada pada pojok kanan bawah. Setelah itu penulis mendefinisikan bahwa pixel yang teridentifikasi sebagai *ridges* dari sidik jari akan diberi nilai 0 dan pixel yang teridentifikasi sebagai *veins* akan diberi nilai 1. Untuk dapat menambah jumlah panjang kunci yang akan di generate pembaca juga dapat menambahkan karakter nilai definisi. Sebagai contoh untuk dua karakter *ridges* dapat didefinisikan sebagai 00 atau 11 dan untuk *veins* dapat didefinisikan sebagai 01 atau 10. Jika yang diambel n karakter maka tingkat kesulitan untuk memecahkan variasi kunci ini juga akan bertambah 2^n sehingga *chipher* teks akan semakin sulit untuk didekripsi. Namun pada makalah ini penulis hanya akan membahas pendefinisian dari komponen sidik jari tersebut dalam hal ini yaitu *ridges* atau *veins* dengan satu karakter saja yaitu 0 dan 1.

Setelah mendapatkan nilai dari pixel- pixelnya, penulis akan menaruh hasilnya pada pada sebuah variable yang sebut saja variable A. untuk nilai dari pixel pada poin (0,0) yang berada pada pojok kiri atas dari file gambar sidik jari yang telah dipilih maka akan dimasukkan nilainya kedalam variable $A_{0.}$ Hal tersebut dilakukan untuk pixel selanjutnya sampai dengan pixel pada posisi pojok kanan atas sehingga akan dihasilkan A_0 sampai dengan $A_{i.}$ Lalu nilai A_0 sampai dengan $A_{i,}$ akan penulis simpan kedalam sebuah variable baru yang penulis beri nama C_0. Untuk baris selanjutnya penulis akan melakukan hal yang sama dari kolom 1 sampai I dan disimpan kembali kedalam A_1 sampai dengan A_i lalu nilainya akan dimasukkan kedalam C_2 dan seterusnya sampai pixel terakhir dan didapatkan C_0 sampai dengan C_j dimana j adalah jumlah baris yang ada. Algoritma sederhananya dapat diberikan sebagai berikut.

```
input(fileImage.bmp);
iniciatePixel(fileImage.bmp);

for(j= 0; j< maxRowPixel; ++j){
    for(i= 0;i< maxColumnPixel; ++i){
        if(getPixel(i,j)=Veins){
            A[i]=1;
        } else{
            A[i]=0;
        }
    }
    C[j]=A;
}
```

Keterangan:

- Fungsi yang pertama, input() adalah fungsi untuk menerima file gambar dari input user.
- Fungsi yang kedua adalah iniciatePixel() adalah fungsi yang masukkannya berupa file gambar dan akan menghasilkan pixel dari gambar tersebut dalam bentuk matriks yang berupa baris dan kolom yang berisi komponen sidik jari tersebut apakah *veins* atau *ridges.*
- Iterasi for yang pertama berguna untuk mengulang dari baris pertama sampai dengan baris maksimal pada file gambar yang diterima.
- Iterasi for yang kedua berguna untuk mengulang dari kolom pertama sampai dengan kolom maksimal pada file gambar.

Lalu tahap kedua yang penulis lakukan adalah meng-*generate* kunci dari file gambar sidik jari pihak penerima. Selanjutnya penulis mengecek satu persatu pixel dari gambar yang telah dipilih atau seleksi dari pixel yang berada di pojok kiri atas sampai pada pixel yang berada pada pojok kanan bawah. Setelah itu penulis mendefinisikan bahwa pixel yang teridentifikasi sebagai *ridges* dari sidik jari akan diberi nilai 0 dan pixel yang teridentifikasi sebagai *veins* akan diberi nilai 1.

Setelah mendapatkan nilai dari pixel- pixelnya, penulis akan menaruh hasilnya pada pada sebuah variable yang sebut saja variable B. untuk nilai dari pixel pada poin (0,0) yang berada pada pojok kiri atas dari file gambar sidik jari yang telah dipilih maka akan dimasukkan nilainya kedalam variable $B_{0.}$ Hal tersebut dilakukan untuk pixel selanjutnya sampai dengan pixel pada posisi pojok kanan atas sehingga akan dihasilkan B_0 sampai dengan $B_{i.}$ Lalu nilai B_0 sampai dengan $B_{i,}$ akan penulis simpan kedalam sebuah variable baru yang penulis beri nama D_0. Untuk baris selanjutnya penulis akan melakukan hal yang sama dari kolom 1 sampai I dan disimpan kembali kedalam B_1 sampai dengan B_i lalu nilainya akan dimasukkan kedalam D_2 dan seterusnya sampai pixel terakhir dan didapatkan D_0 sampai dengan D_j dimana j adalah jumlah baris yang ada. Algoritma sederhananya dapat diberikan sebagai berikut.

```
input(fileImage.bmp);
iniciatePixel(fileImage.bmp);

for(j= 0; j< maxRowPixel; ++j){
    for(i= 0;i< maxColumnPixel; ++i){
        if(getPixel(i,j)=Veins){
            B[i]=1;
        } else{
            B[i]=0;
        }
    }
    D[j]=B;
```

```
}
```

Keterangan:

- Fungsi yang pertama, input() adalah fungsi untuk menerima file gambar dari input user.
- Fungsi yang kedua adalah iniciatePixel() adalah fungsi yang masukkannya berupa file gambar dan akan menghasilkan pixel dari gambar tersebut dalam bentuk matriks yang berupa baris dan kolom yang berisi komponen sidik jari tersebut apakah *veins* atau *ridges.*
- Iterasi for yang pertama berguna untuk mengulang dari baris pertama sampai dengan baris maksimal pada file gambar yang diterima.
- Iterasi for yang kedua berguna untuk mengulang dari kolom pertama sampai dengan kolom maksimal pada file gambar.

Setelah penulis berhasil mendapatkan nilai dari C_0, C_1, C_2, ..., C_j. dan D_0, D_1, D_2, ..., D_j. yang didapat dari sidik jari pengirim dan penerima yang sudah didigitalisasi dalam bentuk file gambar dalam format bitmap, maka penulis akan menggenerate sebuah kunci yang unik berdasarkan nilai tersebut. Kunci yang akan penulis buat didapat dari hasil kombinasi nilai C_0 akan penulis tambahkan dengan nilai 3 bit pertama dari D_0 pada nilai awalnya. Sisa dari bit D_0 yang tersisa dipakai untuk menambahkan bit dari C_0 diakhirnya dan penulis masukkan nilai tersebut pada variabel yang dapat penulis sebut E sehingga dari kombinasi *shifted bit* dari nilai C_0 dan D_0 penulis dapatkan nilai E_0. Hal tersebut penulis iterasi sampai didapatkan nilai dari E_0 sampai dengan E_j. Nilai tersebut sudah dapat penulis jadikan kunci untuk enkripsi pada OTP.

Agar keamanan dari kunci tersebut bertambah maka dari pihak penerima dan pengirim dapat memilih lagi sebuah sub kunci yang besarnya sebesar 128 bit dan nilainya hanya diketahui oleh kedua belah pihak tersebut sehingga walaupun kriptanalis dapat mendapatkan kedua sidik jari yang dijadikan dasar kunci mereka tetap tidak akan mendapatkan semua kuncinya karena ada sub kunci yang mereka tidak ketahui yaitu sebesar 128 bit tersebut yang akan penulis *shifting* nilainya setelah bit pertama pada E_0. Dengan begitu tingkat kesulitan untuk memecahkannya bertambah menjadi 2^128 kalinya dari kunci semula dengan begitu penulis mendapatkan kunci baru yang tingkat kesulitan untuk memecahkannya adalah 2^128.

Setelah itu kunci yang penulis *generate* memiliki besar dalam bit adalah:

Size of Key= 2 x (p x l) + (128 x l)

p : Panjang dari file gambar dalam ukuran pixel
l : lebar dari file gambar dalam ukuran pixel

Dengan kemampuan menggenerate kunci yang yang besar tersebut penulis dapat menggenerate pesan yang cukup panjang. Namun panjang kunci juga terbatas karena sidik jari manusia mempunyai banyak *veins* dan *ridges* yang terbatas. Oleh karena itu prakiraan kunci yang dapat di-*generate* dengan metode ini yaitu tidak lebih dari 2100 karakter. Namun hal kunci yang di-*generate* dapat ditambah kemampuan panjangnya dengan mengganti variabel dari *ridges* dan *veins*-nya tidak dengan satu bit yaitu 0 dan 1 tetapi dengan menggantinya menjadi dua, tiga, empat dan seterusnya sehingga panjang kunci yang dihasilkan dari proses *generate* ini juga dapat bertambah.

Sehingga panjang kunci yang dihasilkannya besarrnya menjadi seperti persamaan berikut:

Size of Key= 2 x (n x p x l) + (128 x l)

p : panjang dari file gambar dalam ukuran pixel
l : lebar dari file gambar dalam ukuran pixel
n : banyak representasi bit pada *veins* atau *ridges*

D. Meng-generate Kunci dengan Menggunakan Komponen Minutiae Point

Setelah penulis meng-*generate* kunci dengan komponen sidik jari yakni *veins* dan *ridges*, sekarang penulis akan memanfaatkan satu lagi komponen dari sidik jari yaitu *minutiae*. Penulis akan meng-*generate* kunci dengan menggunakan algoritma yang berbeda dari yang penulis telah terangkan sebelumnya.

Pada metode dengan menggunakan *minutiae point* ini penulis hanya akan memakai salah satu sidik jari saja. Oleh karena jumlahnya yang berbeda antara manusiannya maka sulit untuk mengkombinasikan antara sidik jari penerima dan pengirim pesan seperti yang penulis lakukan sebelumnya. Oleh karena itu dengan cara ini penulis cukup memakai sidik jari dari salah satu pihak saja yaitu pengirim atau penerima pesan.

Setelah penulis tentukan sidik jari siapa yang akan penulis gunakan maka hal yang pertama yang penulis lakukan adalah mencari semua titik dari *minutiae point*. Setelah mendapatkan kumpulan titik tersebut maka penulis akan mencoba untuk memetakannnya kedalam sebuah matriks berdasarkan jarak antara *minutiae point* tersebut . missal penulis mengambil pojok kiri atas sebagai daerah acuan awal sehingga penulis dapat mnggurutkan *minutiae point* yang penulis dapat sesuai dengan jarak pixelnya antara acuan misalnya untuk *minutiae point* yang paling atas dan berada di paling kiri maka mempunyai nilai *minutiae point* satu dan yang setelahnya adalah dua, dan seterusnya sampai pada anggka jumlah minutiae yang ditemukan pada sidik jari tersebut.

Setelah menentukan urutan tahap selanjutnya adalah menetukan jarak antara *minutiae point* satu dengan yang lainnya. Untuk jarak antara *minutiae point* yang pertama dengan yang kedua penulis akan menaruh nilainya dalam sebuah variabel yang dapat penulis sebut variabel A_{12}, lalu cari jarak antara *minutiae point* yang pertama dengan *minutiae point* yang ketiga dan nilai yang didapatkan akan penulis taruh kembali dalam variabel A_{13} dan seterusnya sampai dengan variabel A_{1m} dimana m adalah jumlah

maksimal dari *minutiae point* yang ditemukan dalam sidik jari tersebut. Jarak antara *minutiae point* satu dengan yang lainnya akan penulis *convert* menjadi bit dengan panjang delapan bit.sebagai contoh jarak antara *minutiae point* yang pertama dan yang kedua adalah empat pixel maka nilai yang penulis isi kedalam matrix tersebut adalah 000000100.

Tahap selanjutnya adalah memilih kunci dengan panjang 128 bit hal ini memeilki tujuan yang sama dengan yang sebelumnya yaitu walaupun kriptanalis mendapatkan sidik jari yang dipakai untuk meng-*generate* kunci namun kriptanalis masih akan tetap kesulitan untuk memecahkan kunci yang 128 bit ini. Selanjutnya sesuai dengan nilai barisnya maka penulis akan melakukan *shifting* pada bit yang sesuai dengan nilai tersebut missal pada A_{1n} dimana n adalah kolom dari matriks tersebut. Maka penulis akan *shifting* bit yang pertama dengan nilai 128 bit itu. Selanjutnya sampai dengan maksimal barisnya di-*shifting* sama dengan barisnya pada tempatnya di dalam matriks. Dibawah ini adalah gambaran dari algoritma untuk meng-*generate* kunci dari *minutiae point* yang ada:

```
  input(fileImage.bmp);
  iniciateMinutiaepoint(fileImage.bmp);

  for(j= 0; j< maxMinutieaePoint; ++j){
      for(i= 0;i< maxMinutieaePoint;
++i){

Aj,i=getDistance(MinutiaePointj,
MinutiaePointi);
      }
  Kj= shiftbit(Aj,I, j, 128bitKey);
  }
```

Keterangan:

- Fungsi yang pertama, input() adalah fungsi untuk menerima file gambar dari input user.
- Fungsi yang kedua adalah `iniciateMinutiaepoint`() adalah fungsi yang masukkannya berupa file gambar dan akan menghasilkan *minutiae poin* –nya dari satu sampai yang terakhir.
- Iterasi for yang pertama berguna untuk mengulang dari baris pertama sampai dengan baris maksimal pada file gambar yang diterima.
- Iterasi for yang kedua berguna untuk mengulang dari kolom pertama sampai dengan kolom maksimal pada file gambar.
- Setelah keluar dari iterasi yang pertama maka nilai hasil dari iterasi pertama akan di-*shift* bit dengan 128 bit key yang telah dipilih sesuai dengan barisnya

Keamanan kunci yang di-*generate* dengan metode ini tidak kalah aman dengan metode sebelumnya dan lebih simpel karena hanya menggunakan dengan satu sidik jari dan kunci yang dihasilkan juga acak dan unik. Oleh karena itu keamanannya tidak penulis ragukan lagi.

Panjang kunci yang dapat di-*generate* dengan metode ini adalah sebsesar:

Size of Key= 2 x (*s* x *s*) + (128 x *s*)

s : sisi dari matriks yang dibentuk oleh *minutiae point* yang bergantung dari jumlah *minutiae point* itu sendiri yang besarnya sama dengan *minutiae point*yang ada dalam sidik jari tersebut.

III. PENGUJIAN KUNCI

A. Panjang Kunci yang Dapat Di-generate

Pada bab sebelumnya penulis sudah dapat melihat panjang kunci yang dapat di-*generate* pada metode yang memanfaatkan *veins* dan *ridges* pada sidik jari dan *minutiae point* pada sidik jari adalah kurang lebih sepenulisr dua ribu seratus karakter dan itu pun masih bisa ditambah lagi panjangnya dengan metode tambahan yang sudah dijelaskan yaitu.

Untuk *veins* dan *ridges:*

Size of Key= 2 x (*n* x *p* x *l*) + (128 x *l*)

p : panjang dari file gambar dalam ukuran pixel
l : lebar dari file gambar dalam ukuran pixel
n : banyak representasi bit pada *veins* atau *ridges*

Untuk *minutiae points:*

Size of Key= 2 x (*s* x *s*) + (128 x *s*)

s : sisi dari matriks yang dibentuk oleh *minutiae point* yang bergantung dari jumlah *minutiae point* itu sendiri yang besarnya sama dengan *minutiae point*yang ada dalam sidik jari tersebut

Sedangkan dengan menggenerate kunci secara konvensional untuk menghasilkan lima puluh karakter sajadengan cara yang tersebut dirasa sangat sulit apalagi untuk menggenerate kunci yang panjangnya ribuan. Selain itu si penerima dan pengirim tidak mungkin dapat meng-*generate* kunci yang sangat yang sama dalam waktu yang berbeda sehingga kunci tersebut juga harus dikirim melalui suatu saluran.

B. Keunikan dan Keacakan Kunci yang Dapat Di-generate

Kunci ysng dihasilkan dengan cara ini tentu saja sangat unik dan acak. Keunikan itu didapat dari sidik jari manusia yang penulis pakai sehingga tidak mungkin ada kunci yang sama yang dihasilkan dari sidik jari yang berbeda. Selain itu keacakannya sudah jelas dengan menggabungkan teknik *shift bit* dan menggabungkannya

dengan letak dari komponen jari manusia seperti *ridges*, *veins*, serta *minutiae points* yang letaknya acak pada setiap manusia.

Bayangkan dengan metode konvensional lainnyayang kunci randomnya kadang- kadang tidak benar random sebagai contoh. Jika penulis memikirkan kata untuk dirandom missal seratus karakter maka mungkin tiga puluh karakter pertama mungkin tersusun secara acak namun pada karakter seterusnya penulis akan memikirkan angka yang kemungkinan berulangnya lebih tinggi karena penulis sudah kehabisan ide.

C.Kefektifan dan Keefisienan

Penulis akan membahas keefektifan dan keefisienan dalam beberapa hal yaitu:

- Biaya
- Transmisi
- Serangan
- Autentikasi

Hal yang pertama yang akan penulis uji adalah biaya. Pada cara konvensional kunci akan di-*generate* oleh salah satu pihak saja. Setelah kunci didapatkan pihak tersebut akan mengirimkan kuncinya kepada pihak lain yang bersangkutan. Hal tersebut dikarenakan tidak mungkin orang yang berbeda dapat meng-*generate* kunci yang random dan unik secara bersamaan. Oleh karena itu kunci dikirim melalui suatu saluran yang sangat aman. Hal ini tentu saja menimbulkan masalah baru karena biasanya saluran yang aman sangat lambat dan mahal sehingga tidak efektif dan efisien dari segi waktu dan biaya.

Namun jika kita bandingkan dengan metode yang penulis buat ini, kita tidak memerlukan saluran khusus untuk mengirmkan kunci. Karena kunci dapat di-*generate* oleh kedua belah pihak, pengirim dan penerima pesan, yang berkepentingan saja. Biaya dan waktu yang akan lebih murah dan cepat seratus persen dibandingkan dengan metode diatas.

Hal yang kedua adalah transmisinya. Pada cara konvensional diperlukan suatu protocol khusus untuk mengirimkan kunci sehingga kunci tidak bisa dilewatkan pada saluran yang memiliki protocol sembarangan (umum/ tidak khusus untuk share message yang sangat private) seperti http, smtp, ftp, dll). Sehingga untuk membangun saluran tersebut diperlukan biaya yang tidak sedikit.

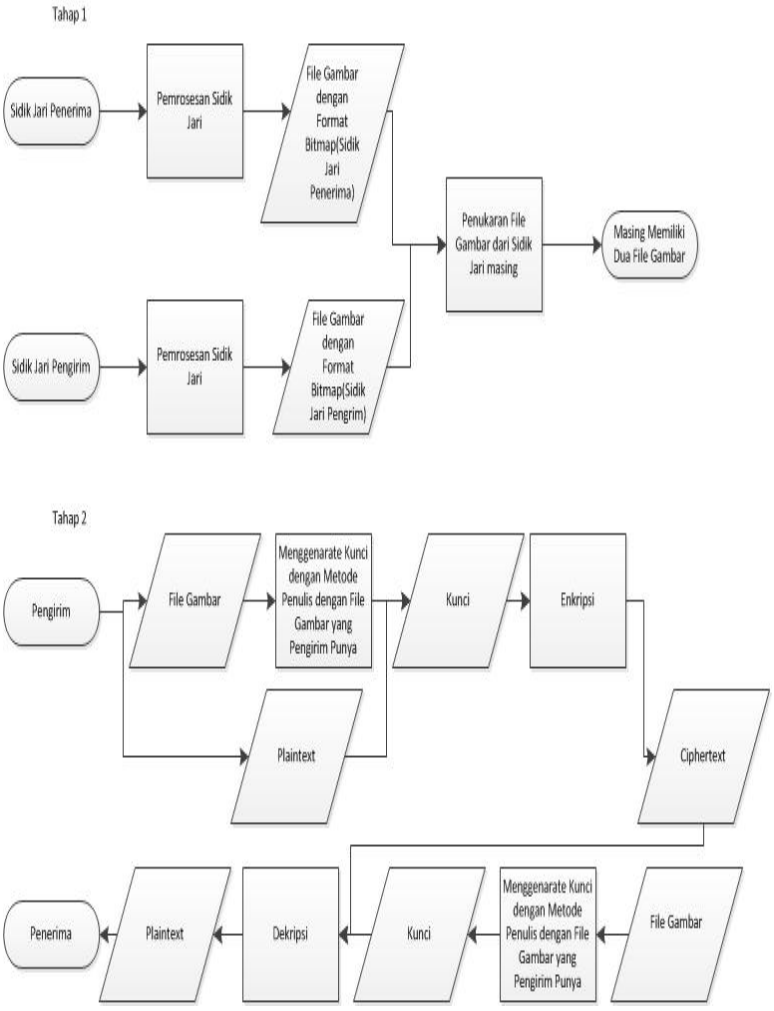
Pada metode penggunaan sidik jari ini, file image yang berisi gambar sidik jari dapat dikirimkan melalui saluran transmisi dengan protocol yang umum semisal http atau smtp karena walaupun kriptanalis mendapatkan file sidik jari tersebut dia tidak akan tahu sub key 128 bit yang dipilih dan bagian mana yang dipakai oleh gambar tersebut untuk menggenerate kunci. Oleh karena itu mustahil bagi kriptanalis untuk mendapatkan *plaintext* dari *ciphertext* yang didapat.

Serangan yang dilakukan untuk memecahkan kuncinya untuk *One Time Pad Cipher* adalah mustahil karena panjang kunci memiliki panjang yang sama dengan panjang *plaintext*-nya sehingga mustahil bagi kriptanalis untuk mengetahui kunci. Jika kriptanalis ingin mengambil sidik jari yang berupa file image dengan format bitmap hal ini juga amat mustahil walapun kriptanalis mendapatkannya namun ada 128 bit key yang tidak diketahui olehnya dan bagian pixel mana sampai pixel mana yang diambil untuk meng-*generate* juga tidak diketahui olehnya.

Untuk autentikasi jelas setiap sidik jari berbeda antara satu dengan lain orang oleh karena itu autentikasi terhadap si penerima pesan sudah pasti terjamin. Berbeda dengan cara konvensioanal penerima pesan akan terlebih dahulu menerima kunci dari si pengirim pesan baru dapat mengeautentikasi sebagai penerima yang menurut penulis cara ini amat tidak efektif.

D. Alur Kerja

Dibawah ini adalah alur kerja dari peng-*generate*-an kunci dengan memanfaatkan komponen *veins* dan *ridges:*



Gambar 2. Diagram Alir dari Metode *generate* Kunci dengan *Veins* dan *Ridges*

Diagram diatas menunjukan bagaimana proses si penerima dan pengirim surat dari mulai menggenerate kuni sampai dengan mengirim dan menerima pesan serta mengekstrak pesan tersebut dengan menggunakan kunci yang telah di-*generate* dengan metode yang telah penulis berikan.

Setelah kunci sudah didapatkan dengan metode ini maka langkah selanjutnya adalah mengguanakan kunci tersebut untuk diterapkan dalam enkripsi dengan menggunakan metode *One Time Pad Cipher*.

IV. KESIMPULAN

Pada makalah ini penulis membahas tentang pemanfaatan *biometrics* dalam hal ini adalah sidik jari manusia untuk meng-*generate* kunci yang akan dipakai untuk mengenkripsi pesan dengan metode enkripsi *One Time Pad Cipher*. Dengan menggunakan metode ini ada beberapa keuntungan yang telah diuji pada bab sebelumnya diatas diantaranya:

- Biaya
- Transmisi
- Serangan
- Autentikasi

Dengan parameter tersebut sudah terbukti dengan menggunakan pemanfaatan sidik jari dalam peng-*generate*-an kunci lebih efisien dan efektif dari pada cara konvensional sehingga ketika kita memanfaatkan algoritma *One Time Pad Cipher* untuk mengirimkan pesan. Kita tidak perlu khawatir lagi dengan bagaimana persoaalan dalam menggenerata kuncinya yang biasanya menjadi masalah seiiring dengan panjangnya pesan yang akan dikirim

V. SARAN

Selain itu aplikasi untuk meng-*generate* ini tidak hanya terbatas hanya pada sidik jari saja tetapi kita dapat menambahakannya dengan komponen biometrics yang lainnya seperti retina atau DNA dan lain. Sehingga tingkat keamanan dalam meng-*generate* kuncinya lebih aman lagi.

REFERENCES

[1] Munir, Rinaldi, Diktat Kuliah IF5054 Kriptografi, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, 2006.

[2] Andrew S. Tanenbaum, “Computer Networks”, New Jersey, Prentice Hall, 2001.

[3] http://www.cse.msu.edu/biometrics/fingerprint.html tanggal akses 16 maret 2011.

[4] http://www.gizmag.com/nec-develops-contactless-fingerprint-scanner/17989/ tanggal akses16 maret 2011.

[5] Paul, Reid “Biometrics and Network Security”, New Delhi Pearson Educational Series, 2004.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 14 April 2011

Nama dan NIM