DESAIN DAN ANALISIS PROTOKOL PENGIRIMAN DATA TRANSFORMASI SIDIK JARI MENGGUNAKAN PEWARNAAN GRAF

Doni S. Pambudi[1)] dan Tohari Ahmad[2)]
[1, 2)] Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia
e-mail: donisp06@gmail.com[1)], tohari@if.its.ac.id[2)]

ABSTRAK

Sidik jari merupakan salah satu biometrik yang digunakan secara luas untuk otentikasi pengguna. Karakteristik sidik jari adalah permanen dan melekat kepada pengguna sehingga data sidik jari tidak boleh diketahui oleh orang lain. Salah satu metode pengamanan data sidik jari yaitu dengan melakukan transformasi pada fitur sidik jari, transformasi menghasilkan data lain yang tidak dapat dikembalikan ke bentuk asal, kemudian data dikirimkan ke server untuk dicocokkan dengan basis data. Transfer data menimbulkan celah keamanan bocornya data hasil transformasi sehingga dimungkinkan dapat dianalisa dan disusun kembali menjadi data fitur sidik jari. Pada penelitian ini diusulkan sebuah protokol untuk mengamankan pengiriman data hasil transformasi sidik jari dengan menggunakan pewarnaan graf pada proses enkripsi dan dekripsi. Protokol ini diawali dengan pertukaran bilangan kunci antara server dengan client, bilangan kunci yang disepakati digunakan untuk mengacak IV, selanjutnya proses enkripsi dan deskripsi menggunakan graf yang telah diwarnai dengan IV teracak. Dari hasil analisis dan percobaan yang dilakukan, protokol yang diusulkan mampu mencapai kecepatan enkripsi 0.0160 milidetik dan dekripsi 0.0264 milidetik untuk data sebesar 512 byte dengan peluang maksimum data dapat dipecahkan sebesar $3.719933267899012 \times 10^{41}$.

***Kata Kunci**: Pewarnaan Graf, Protokol Pengiriman Data, Transformasi Sidik Jari.*

ABSTRACT

Fingerprints are one of the widely used biometrics for user authentication. Fingerprint characteristics are permanent and attached to user so that the fingerprint data should not be known by others. One method of securing the fingerprint data by performing transformations on fingerprint features, the transformation generates other data cannot be restored to the original form, then the data is sent to the server to be matched with the database. Data transmission may have certain security risk which any third person may expose transformed data. The exposed information may be further analyzed and re-compiled so that the original fingerprint data can be obtained. In this research proposed a protocol for secure data transmission fingerprint transformation results using graph coloring in the process of encryption and decryption. This protocol started by the key exchange between client and server, determined the key number used to scramble IV, furthermore encryption and decryption process utilize graph that has been colored with scrambled IV. Both analysis and experiment are performed on proposed protocol which is able to encrypt and decrypt 512 byte data in 0.016 milliseconds and 0.0264 milliseconds, respectively, with maximum 3.719×10^{41} probabilities of breakage.

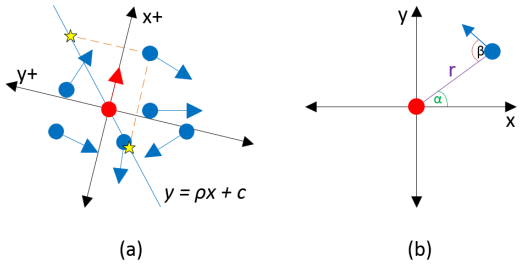
***Keywords:** Data Transmission Protocol, Fingerprint Transformation, Graph Coloring.*

I. PENDAHULUAN

OTENTIKASI diperlukan untuk membuktikan bahwa pengguna berhak mengakses data atau layanan tertentu [1]. Biometrik telah digunakan secara luas sebagai salah satu cara otentikasi pengguna [2], keunggulan biometrik yaitu sulit digandakan, permanen, dan melekat pada pengguna. Karena sifat tersebut data asli biometrik tidak boleh bocor kepada pihak lain karena dapat disusun kembali dan digunakan menjadi alat otentikasi oleh pihak yang tidak bertanggungjawab, otentikasi dapat dilakukan selamanya karena biometrik membawa sifat permanen [3]. Sidik jari adalah contoh dari biometrik yang telah dikenal dan digunakan secara luas pada berbagai bidang [7].

Salah satu cara melindungi data biometrik dikenal dengan konsep *cancelable template*, konsep ini menyimpan data hasil transformasi yang disebut *template*, proses pencocokan dilakukan antara hasil transformasi sehingga data asli tidak lagi dibutuhkan, ketika *template* bocor maka dibuat ulang dengan menggunakan parameter atau kunci yang berbeda [6]. Transformasi yang dilakukan dapat berupa transformasi *cartesian*, transformasi *polar* atau transformasi fungsional [6].

Penelitian sebelumnya pada sidik jari, transformasi *cartesian* dilakukan dengan membuat garis proyeksi kemudian titik *minutiae* diproyeksikan ke garis tersebut secara vertikal dan horizontal, garis proyeksi dibagi menjadi beberapa partisi sehingga titik hasil proyeksi terkelompok ke dalam partisi tertentu, jumlah titik dalam sebuah partisi membentuk vektor hasil transformasinya [4]. Transformasi polar dilakukan pada penelitian [5] dengan mencari hubungan antara titik *minutiae*, hubungan ini di transformasi dengan mengubah sektor berdasarkan sudut putar tertentu dan mengubah jarak antara titik



Gambar 1. (a) Metode garis proyeksi; (b) Metode *pair-polar*

minutiae acuan dengan *minutiae* tetangga. Transformasi fungsional diusulkan pada penelitian [11], hubungan antara titik *minutiae* diubah dengan fungsi polinomial kemudian ditambahkan titik *minutiae* palsu agar data yang disimpan tidak mudah diketahui data yang sebenarnya.

Setiap jenis transformasi menghasilkan vektor hasil yang akan dibandingkan pada proses pencocokan. Proses pencocokan pada perkembangannya tidak lagi dilakukan pada satu komputer yang sama, namun mulai diimplementasikan dengan model *client-server*. Model ini memiliki keunggulan yaitu kemudahan pemeliharaan basis data dan proses pencocokan yang terpusat sehingga mampu mengakomodir banyak *client* sekaligus. Proses pencocokan dilakukan dengan mengirimkan data hasil transformasi melalui jaringan, pengiriman ini menimbulkan celah data dapat disadap dan dianalisa kemudian disusun kembali menjadi data sidik jari asli maupun buatan sebagai media otentikasi yang valid. Proses melindungi pengiriman data hasil transformasi sidik jari telah diteliti pada [10] dengan menggunakan *certificate authority* pada proses pertukaran data, sebelum dikirimkan ditambah data acak untuk mengaburkan data sebenarnya.

Penelitian yang telah dilakukan sebelumnya fokus pada penggunaan metode enkripsi dan dekripsi yang telah ada dengan penggunaan kunci statis yang telah ditentukan, pada penelitian ini diusulkan protokol yang mampu melakukan pertukaran data dengan kunci dinamis sehingga keluaran dari proses enkripsi dan dekripsi sulit dianalisa karena tidak adanya pola keterkaitan antara data yang dikirim dengan data yang dikirimkan selanjutnya. Pembahasan pada penelitian ini meliputi metode pertukaran kunci acak antara *client* dengan *server*, metode pengacakan IV berdasarkan kunci acak, metode pewarnaan graf, metode enkripsi, dan metode dekripsi.

II. PENELITIAN TERKAIT

Pada bagian ini dijelaskan mengenai penelitian terkait yang menjadi referensi pada protokol yang diusulkan.

A. *Transformasi Fitur Sidik Jari*

Proses transformasi pada fitur sidik jari dilakukan dengan mengubah data fitur menjadi data lain yang tidak bisa dikembalikan dengan membalikkan proses yang sama, dengan kata lain A + B = C tidak sama dengan C – B = A. Proses transformasi menurut [6] terdiri dari 3 jenis yaitu transformasi cartesian, transformasi polar, dan transfomasi fungsional.

Metode garis proyeksi adalah contoh transformasi cartesian, metode ini membuat garis proyeksi dengan persamaan y = ρx + c dengan ρ adalah kemiringan garis dan c adalah konstanta seperti terlihat pada Gambar 1a [4]. Semua titik *minutiae* diproyeksikan ke dalam garis tersebut dengan menghitung nilai koordinat relatif titik *minutiae* terhadap titik pusat (1), kemudian mencari koordinat titik proyeksi horizontal (2) dan proyeksi vertikal (3). Vektor hasil tranformasi garis proyeksi adalah jumlah titik proyeksi di setiap partisi, misalkan v = {2,4,0,5,6,4}.

$$\begin{bmatrix} x_{rel} \\ y_{rel} \end{bmatrix} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (1)$$

$$(x_h, y_h) = \left(\frac{y_{rel}-c}{\rho}, y_{rel}\right) \quad (2)$$

$$(x_v, y_v) = (x_{rel}, \rho x_{rel} + c) \quad (3)$$

Metode *pair-polar* adalah contoh transformasi polar, metode ini menggunakan hubungan antara titik *minutiae* dengan mencari {r, α, β}, nilai hubungan ditunjukkan pada Gambar 1b. Transformasi dilakukan dengan mengubah *sector* sehingga nilai α dan β berubah (4) dan mengubah nilai r (5). Transformasi *pair-polar* menghasilkan vektor hasil yaitu {{r_1, α_1, β_1},{r_2, α_2, β_2},…{r_n, α_n, β_n}}.

$$newsector = abs(oldsector * v_w)\ mod(totalsector) \quad (4)$$

$$r_{ij} = (r_{ij} * r_w)\ mod\ (\mu)/r_w \quad (5)$$

B. *Algoritma Diffie-Hellman*

Algoritma diffie-hellman (DH) digunakan untuk pertukaran kunci enkripsi melalui media terbuka yang tidak aman. DH merupakan salah satu contoh algoritma untuk metode pertukaran kunci yang paling awal [9]. DH menggunakan bilangan prima (p) dan *primitive root modulo* (g) dari bilangan prima yang dipilih. Misalkan diketahui p = 19, g = 2, maka langkah-langkah perhitungan algoritma DH adalah sebagai berikut:

1. *Client* memilih bilangan acak a, misalkan a = 6, *client* mengirimkan nilai A = g^a mod p ke *server*, nilai yang dikirimkan *client* adalah A = 2^6 mod 19, A = 7.

2. *Server* menerima nilai A dari *client*, kemudian memilih bilangan acak b, misalkan b = 14, *server* membalas mengirimkan nilai $B = g^b \bmod p$ ke *client*, nilai yang dikirimkan *server* adalah $B = 2^{14} \bmod 19$, B = 6.
3. *Client* menerima B dari *server*, kemudian *client* menghitung nilai bilangan kunci $s = B^a \bmod p$, $s = 6^6 \bmod 19$, s = 11.
4. *Sever* menerima nilai A dari *client* kemudian *server* menghitung nilai bilangan kunci $s = A^b \bmod p$, $s = 7^{14} \bmod 19$, s = 11.

Dari perhitungan diatas didapatkan bilangan kunci s = 11.

Gambar 2a menunjukkan bahwa pada model otentikasi client-server, data hasil transformasi dikirimkan melalui jaringan kemudian dicocokkan pada basis data template dan dikembalikan nilai cocok atau tidak. Data yang tidak terlindungi pada jaringan dapat dianalisa sehingga menimbulkan celah keamaan data bisa dikembalikan ke bentuk asal atau dibuat data palsunya. Keamanan hasil transformasi ditambahkan dengan melakukan enkripsi ketika data akan dikirimkan ke server terlihat pada Gambar 2b.

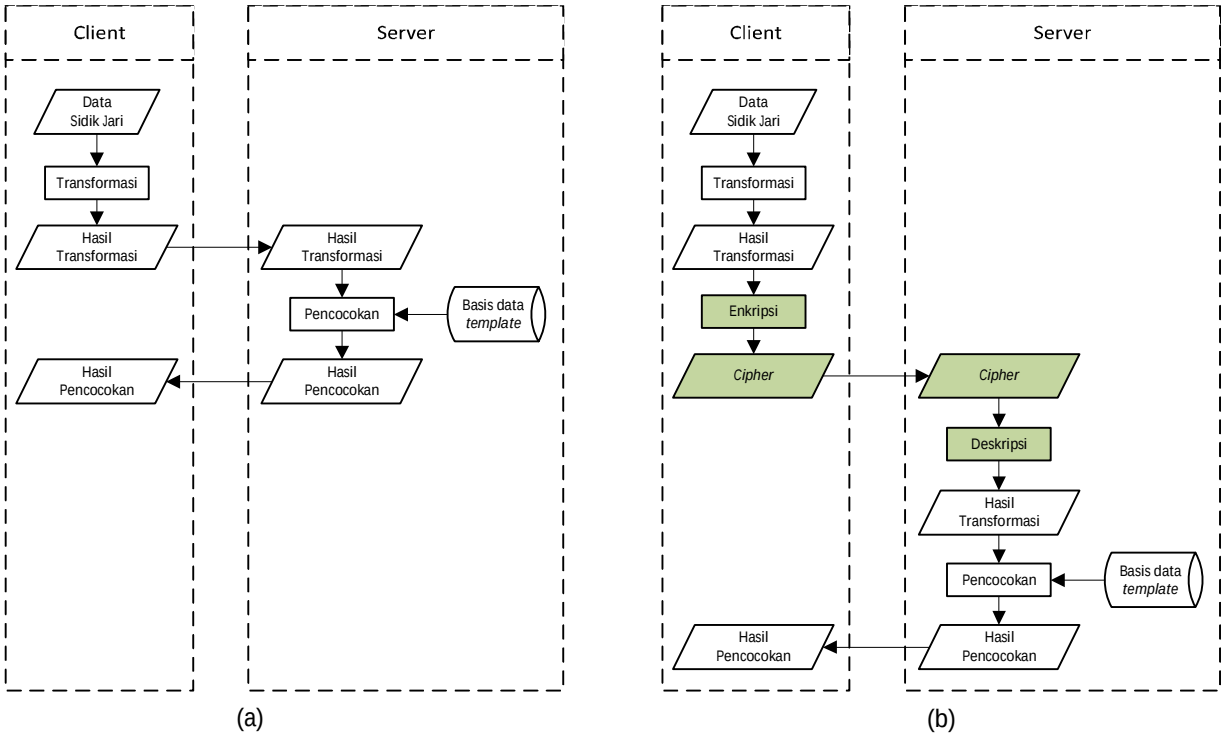
III. METODE YANG DIUSULKAN

Penelitian ini mengusulkan sebuah protokol keamanan pengiriman data hasil transformasi sidik jari dengan melakukan enkripsi sebelum data dikirimkan melalui jaringan. Langkah-langkah protokol yang diusulkan pada Gambar 3 adalah sebagai berikut:

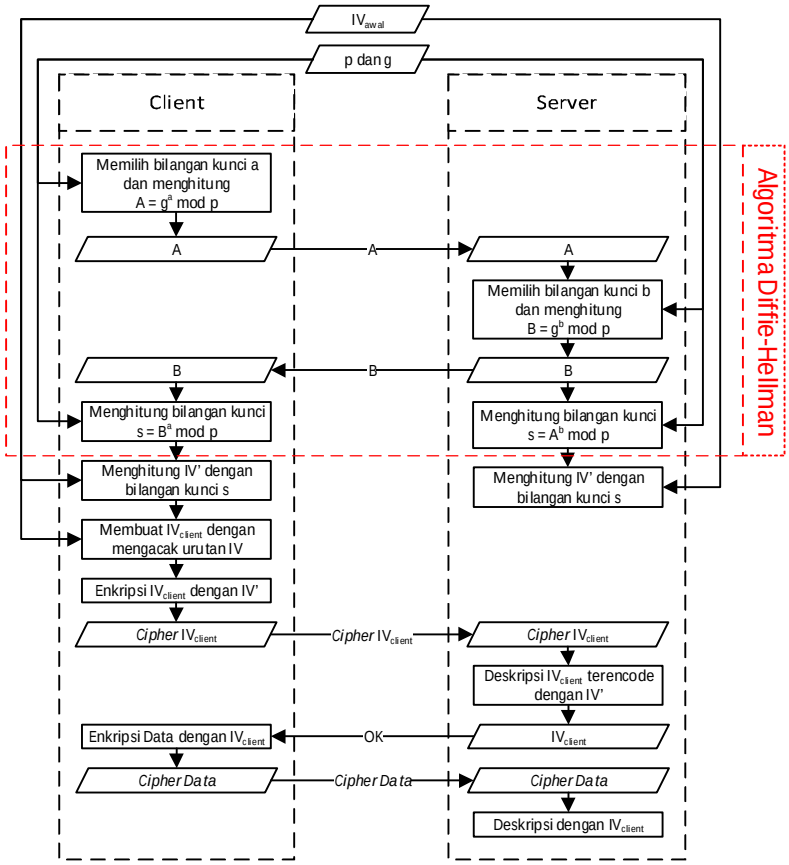
1. Pertukaran bilangan kunci antara *client* dan *server* menggunakan algoritma DH, algoritma DH dapat digantikan oleh algoritma apapun untuk pertukaran kunci publik.
2. Pembuatan IV' untuk enkripsi IV_{client}, IV' dibuat dengan mengacak IV_{awal} menggunakan bilangan kunci yang diperoleh pada langkah pertama.
3. IV_{client} dibuat dengan mengacak susunan IV_{awal}, IV_{client} dikirimkan ke *server* setelah dienkripsi menggunakan IV'.
4. Data hasil transformasi dienkripsi menggunakan IV_{client} dan dikirimkan ke *server*.
5. Server melakukan dekripsi data hasil transformasi kemudian melanjutkan ke langkah selanjutnya mencari *template* yang cocok dengan data yang dikirimkan.

Protokol ini memerlukan 3 buah informasi yang sama antara *client* dan *server*, informasi ini digunakan untuk proses dasar enkripsi, dekripsi, dan pertukaran bilangan kunci, informasi tersebut yaitu:

1. IV_{awal} adalah IV yang disepakati bersama antara *server* dan seluruh *client*, IV ini digunakan sebagai patokan pembuatan IV' berdasarkan bilangan kunci yang dipertukarkan.
2. p adalah bilangan prima yang digunakan oleh algoritma DH.
3. g adalah *primitive root modulo* dari p yang digunakan oleh algoritma DH.



Gambar 2. Diagram alur otentikasi hasil transformasi (a) Tanpa pengamanan data; (b) Dengan pengamanan data



Gambar 3. Diagram alur protokol yang diusulkan

A. *Data yang Dikirim*

Data yang dikirim pada penelitian ini dibatasi angka (0-9) dan 3 karakter pemisah, namun data yang dikirim dapat berupa karakter apapun. Contoh format data yang dikirim dapat dilihat pada Gambar 4, pemisah terdiri dari 3 jenis yaitu: pemisah data vektor v, pemisah angka desimal, dan pemisah data minutiae. Pada penelitian ini pemisah antara data vektor v adalah koma (,), pemisah bilangan desimal adalah titik (.), sedangkan pemisah antar data minutiae adalah titik koma (;).

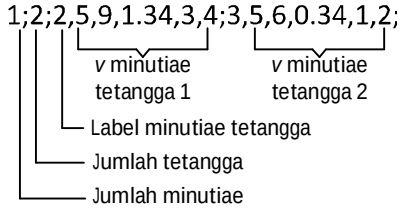
Pada pembatasan data hanya angka, maka untuk pengiriman IV_{client} dan data yang bukan angka harus dikonversi menjadi data angka terlebih dahulu atau meningkatkan data yang dapat dikirim dengan angka dan huruf atau karakter lainnya, jumlah IV yang digunakan harus memenuhi (6). Proses pengubahan menjadi angka salah satunya dilakukan dengan mengubah data menggunakan IV_{awal}, data baru diperoleh dari nilai *index* data lama pada IV_{awal} seperti pada (7).

$$panjang\ IV = panjang\ karakter + panjang\ pemisah \quad (6)$$

$$data\ Baru = index\ data\ lama\ pada\ IV_{awal} \quad (7)$$

Contoh pengubahan data selain angka menggunakan IV_{awal} adalah proses enkripsi IV_{client}, langkah pengubahan data sebagai berikut:

- IV_{awal} = {a,b,c,d,e}
- IV_{client} = {e,c,d,b,a}
- Data yang dienkripsi = {5,3,4,2,1}



Gambar 4. Contoh format data yang dikirimkan

B. *Initialization Vector*

Agar protokol dapat berjalan maka dibutuhkan konfigurasi awal yang berupa *initialization vector* (IV), fungsi utama IV digunakan untuk menandai garis yang menghubungkan antara simpul pada proses pewarnaan graf. Pada penelitian ini IV

adalah kombinasi dari huruf dan angka (a-z dan 0-9), IV tidak terbatas pada huruf dan angka saja dan dapat berupa karakter apapun. IV pada protokol ini terdiri dari 3 yaitu:

1. IV_{awal} yaitu IV yang karakternya terurut atau telah ditentukan sebelumnya dan tidak berubah, IV ini digunakan untuk menghasilkan IV'. Contoh IV_{awal} = {0,1,2,3,4,5,6,7,8,9,a,b,c,d, … , x, y, z}.
2. IV' yaitu IV_{awal} yang diacak dengan bilangan kunci s dan digunakan untuk mengirimkan IV_{client} ke server.
3. IV_{client} yaitu IV yang urutan karakternya acak, IV ini dibuat untuk enkripsi tiap sekali pengiriman data. Contoh IV_{client} = {b,5,3,d,1,t,y,w,x,q, …}

C. Bilangan Kunci

Komunikasi antara *client* dengan *server* dijalankan menggunakan IV yang dimiliki oleh *client* (IV_{client}), tiap komunikasi dimungkinkan menggunakan IV_{client} yang berbeda, IV_{client} diperoleh *client* dengan mengacak urutan dari IV_{awal}, IV_{client} kemudian dikirimkan ke *server* dengan cara melakukan enkripsi menggunakan IV' yang disusun menggunakan bilangan kunci s yang telah disepakati oleh *client* dan *server* menggunakan algoritma DH.

Pertukaran bilangan kunci dapat digunakan tidak hanya untuk sebuah bilangan kunci, namun dapat digunakan untuk menukarkan banyak bilangan kunci sekaligus, untuk banyak bilangan kunci data yang dikirimkan sesuai dengan format (8).

$$S = s_1, s_2, s_3, \dots, s_n \tag{8}$$

D. Memperoleh IV'

Kunci s yang telah disebutkan sebelumnya digunakan untuk mengacak urutan IV_{awal}, hasil acak (IV') inilah yang dipakai menandai garis antara simpul graf, nilai IV_{client} yang akan dikirimkan dari *client* ke *server*. Untuk memperoleh IV' digunakan Algoritma 1.

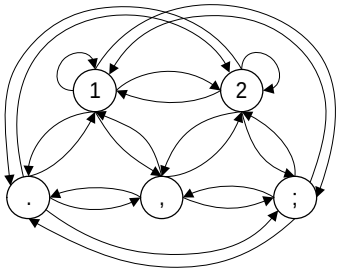
```
IV = array (26 huruf + 10 angka, a-z + 0-9)
i = s mod length(IV)
count = 1
while count < length(IV)
      if IV[i] telah digunakan pada IV'
            i  = ( i + 1 ) mod length(IV)
      IV'[count++] = IV[i]
      i = (i + s) mod length(IV)
```

Algoritma 1. Pengacakan IV menggunakan kunci s

Misalkan:
IV = {1,2,3,4}, s = 2
count = 1, IV' = {2}
count = 2, IV' = {2, 4}
count = 3, IV' = {2,4,3}
count = 4, IV' = {2,4,3,1}

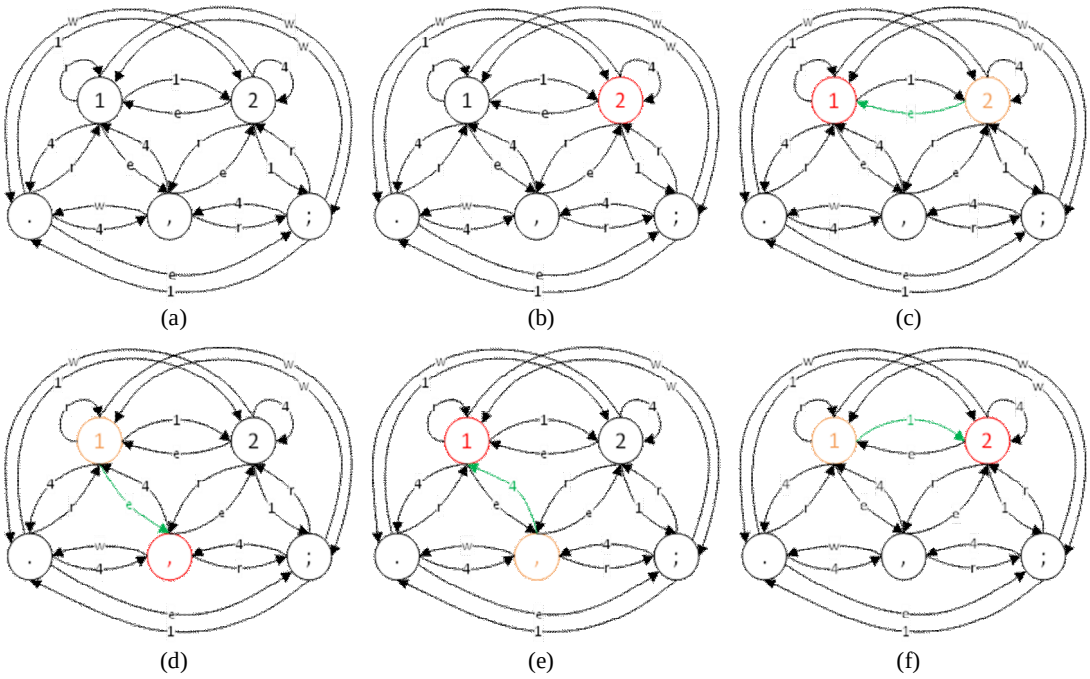
E. Pembuatan dan Pewarnaan Graf

Pada penelitian ini graf terdiri dari 13 simpul yaitu 10 angka (0-9) dan 3 tanda (koma, titik, dan titik koma). Graf terhubung secara langsung dan terarah, seluruh simpul saling terhubung. Simpul angka mempunyai *loop* untuk mengakomodasi angka sama yang berurutan, misal 11, 22, 33, dan lain-lain. Setiap garis ditandai dengan menggunakan IV (IV' jika graf untuk pengiriman IV_{client}, ditandai dengan IV_{client} jika graf untuk pengiriman data transformasi). Contoh graf dapat dilihat pada



Gambar 5. Contoh graf dengan 2 simpul angka

Gambar 5.



Gambar 6. (a) Graf yang telah ditandai IV; (b) Pengambilan titik acuan; (c) Enkripsi data pertama; (d) Enkripsi data kedua; (e) Enkripsi data ketiga; (f) Enkripsi data keempat.

Pewarnaan garis pada graf menggunakan IV yang telah ditentukan, baik IV' maupun IV_{client}. Pada protokol ini pewarnaan dilakukan dengan mengambil tiap IV sesuai dengan jarak sejauh kunci bilangan, jika IV telah digunakan maka geser 1 IV sampai ditemukan IV yang belum digunakan, jika semua IV telah digunakan, set IV menjadi tidak digunakan, ulangi pengambilan IV sampai semua garis tertandai. Pengambilan IV dengan perbedaan jarak sejauh kunci bilangan s memastikan semua nilai pada IV digunakan dalam penandaan pada graf yang dibuat, semakin banyak nilai IV yang digunakan untuk menandai graf, semakin tinggi tingkat keamanan hasil enkripsi dapat dipecahkan.

F. Enkripsi Data

Untuk melakukan enkripsi data masukan maka karakter per karakter diubah menggunakan graf yang telah dibuat dengan langkah sebelumnya. Proses enkripsi data transformasi maupun IV_{client} adalah sebagai berikut:

1. Ambil simpul acuan dengan nilai simpul acuan (s') dihitung menggunakan (9).
2. Ambil sebuah data, dari simpul awal bergerak ke arah simpul dengan nilai simpul sama dengan nilai data yang diambil, catat tanda pada baris sebagai hasil enkripsi.
3. Ganti simpul sekarang dengan simpul tujuan pada langkah 2.
4. Ulangi langkah 2-3 sampai semua data terenkripsi.

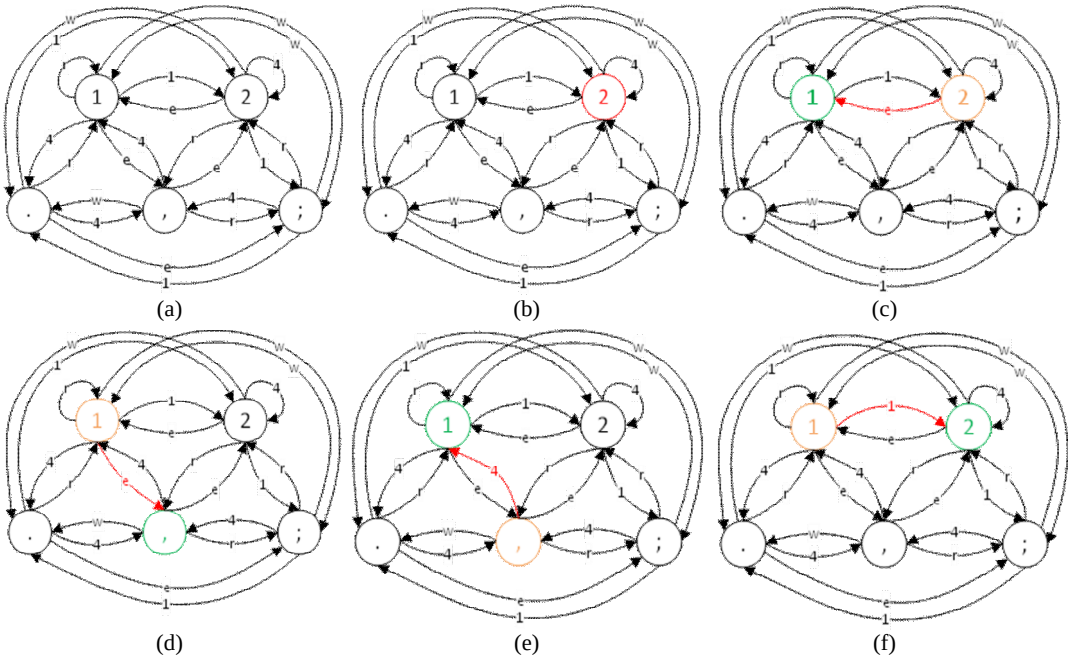
$$s' = s \ mod \ (jumlah \ simpul \ selain \ tanda) \tag{9}$$

Contoh, diketahui:

- IVclient = {I,4,e,w,r}.
- Data masukan = {1,12}.
- Kunci bilangan s = 2.
- Graf telah dibuat dan ditandai dengan IV_{client} terlihat pada Gambar 6a.

Langkah enkripsi sebagai berikut:

1. Kunci bilangan s = 2, hitung s' = s mod(10), s' = 2. Nilai s' ini dijadikan acuan dari simpul mana proses enkripsi dilakukan. Proses enkripsi dimulai dari simpul 2 seperti yang terlihat pada Gambar 6b.
2. Bergerak ke arah simpul sesuai dengan karakter pertama pada data masukan. Karakter pertama data adalah 1, maka bergerak menuju ke simpul dengan nomor 1 seperti pada Gambar 6c, diperoleh garis menuju simpul 1 adalah e, maka catat hasil enkripsi untuk karakter pertama adalah e, sehingga hasil enkripsi sementara = e.
3. Data kedua adalah koma, bergerak ke arah koma seperti Gambar 6d, maka hasil enkripsi data kedua adalah e, sehingga hasil enkripsi sementara = ee.



Gambar 7. (a) Graf yang telah ditandai IV; (b) Pengambilan titik acuan; (c) Dekripsi data pertama; (d) Dekripsi data kedua; (e) Dekripsi data ketiga; (f) Dekripsi data keempat.

4. Data ketiga adalah 1, bergerak ke arah 1 seperti Gambar 6e, maka hasil enkripsi data ketiga adalah 4, sehingga hasil enkripsi sementara = ee4.
5. Data keempat adalah 2, bergerak ke arah 2 seperti Gambar 6f, maka hasil enkripsi data keempat adalah 1, sehingga hasil enkripsi = ee41.

Dari proses diatas didapatkan hasil enkripsi dari data {1,12} adalah {ee41}, data **ee41** inilah yang dikirimkan ke *server*.

G. Dekripsi Data

Untuk melakukan dekripsi data masukan maka karakter per karakter diubah menggunakan graf yang telah dibuat dengan langkah sebelumnya. Proses dekripsi data transformasi maupun IV_{client} adalah sebagai berikut:

1. Ambil simpul acuan dengan nilai simpul acuan (s') dihitung menggunakan (9).
2. Ambil sebuah data, dari simpul awal bergerak ke arah simpul dengan nilai garis sama dengan nilai data yang diambil, catat tanda pada simpul sebagai hasil enkripsi.
3. Ganti simpul sekarang dengan simpul tujuan pada langkah 2.
4. Ulangi langkah 2-3 sampai semua data terenkripsi.

Contoh, diketahui:

- IVclient = {I,4,e,w,r}.
- Data masukan = {ee41}.
- Kunci bilangan s = 2.
- Graf telah dibuat dan ditandai dengan IV_{client} terlihat pada Gambar 7a.

Langkah dekripsi sebagai berikut:

1. Kunci bilangan s = 2, hitung s' = s mod(10), s' = 2. Nilai s' ini dijadikan acuan dari simpul mana proses dekripsi dilakukan. Proses dekripsi dimulai dari simpul 2 seperti yang terlihat pada Gambar 7b.
2. Data pertama adalah e, bergerak ke arah simpul dengan garis bertanda e seperti pada Gambar 7c, maka didapatkan hasil dekripsi 1, sehingga hasil dekripsi sementara = 1.
3. Data kedua adalah e, bergerak ke arah simpul dengan garis bertanda e seperti pada Gambar 7d, maka didapatkan hasil dekripsi (,), sehingga hasil dekripsi sementara = 1,.
4. Data ketiga adalah 4, bergerak ke arah simpul dengan garis bertanda 4 seperti pada Gambar 7e, maka didapatkan hasil dekripsi 1, sehingga hasil dekripsi sementara = 1,1.
5. Data keempat adalah 1, bergerak ke arah simpul dengan garis bertanda 1 seperti pada Gambar 7f, maka didapatkan hasil dekripsi 2, sehingga hasil dekripsi = 1,12.

Dari proses diatas didapatkan hasil dekripsi dari data {ee41} adalah {1,12}, data **1,12** inilah yang diproses *server*.

TABEL I
HASIL UJI COBA WAKTU ENKRIPSI DAN DEKRIPSI

Panjang Data (byte)	Waktu Enkripsi (milidetik)	Waktu Dekripsi (milidetik)
512	0.0160	0.0263
1024	0.0314	0.0417
2048	0.0594	0.0880
3072	0.0858	0.1332
4096	0.1153	0.1775
5120	0.1429	0.2203
6144	0.1707	0.2638
7168	0.1970	0.3109
8192	0.2254	0.3513
9216	0.2525	0.3956
10240	0.2805	0.4428
102400	2.8227	4.4871

IV. UJI COBA DAN ANALISIS

Uji coba dilakukan dengan mengukur waktu enkripsi dan dekripsi pada data masukan dengan panjang data yang berbeda, data masukan dibuat secara acak, proses uji coba dilakukan dengan melakukan enkripsi terhadap data masukan, selanjutnya hasil enkripsi dilakukan dekripsi, hasil dekripsi dicocokkan dengan data masukan untuk melakukan pengecekan kebenaran hasil enkripsi dan dekripsi. Setiap panjang data dilakukan percobaan diatas sebanyak 200 kali dan dihitung waktu rata-rata tiap percobaan. Hasil uji coba yang ditunjukkan pada Tabel I kecepatan enkripsi pada panjang data 512 byte adalah 0.0160 milidetik sedangkan kecepatan dekripsi 0.0263 milidetik. Kecepatan enkripsi dan dekripsi berbanding lurus dengan ukuran data yang diproses.

Dari pembahasan diatas panjang IV minimal agar dapat digunakan untuk menandai semua hubungan graf sebanyak L = n, dimana L adalah panjang minimal IV, n adalah jumlah simpul yang digunakan dalam graf. Simpul yang digunakan dalam graf tidak terbatas pada contoh diatas, simpul dapat berupa karakter apapun dengan syarat L = n tetap dapat dipenuhi. Bilangan kunci s dalam penjelasan diatas adalah berupa sebuah bilangan, namun dapat juga berupa sekumpulan bilangan, semakin banyak tentunya keamanan semakin meningkat, namun transfer data untuk pertukaran bilangan s juga semakin besar.

Algoritma ini dapat diserang dari 2 sisi yaitu pertama mencoba mendapatkan bilangan kunci s, kemudian memecahkan pengiriman data dengan bilangan kunci tersebut karena telah mengetahui nilai IV', kedua mencoba memecahkan nilai IV_{client}. Kedua teknik serangan diatas telah diantisipasi dengan penggunaan s dan IVclient hanya sekali untuk tiap transaksi. Algoritma DH tidak mengikat untuk diterapkan ke dalam protokol ini, algoritma DH hanya digunakan untuk pertukaran kunci, algoritma sejenis juga dapat diterapkan ke dalam protokol ini. Kunci s tidak dianalisa celah keamanannya karena algoritma pertukaran data tidak mengikat pada DH. Peluang pemecahan IV_{client} adalah sebesar n! dimana n adalah panjang IV_{client}, panjang IV minimal yang digunakan dalam protokol ini adalah sepanjang 36 karakter, sehingga peluang maksimal IV_{client} dapat diketahui adalah $36! = 3.719933267899012 \times 10^{41}$.

V. KESIMPULAN

Keamanan pada data biometrik terus dikembangkan dengan berbagai metode transformasi, otentikasi data hasil transformasi yang dilakukan di *server* memerlukan pengamanan pada proses pengiriman datanya. Pada penelitian ini diusulkan protokol pertukaran data hasil transformasi yang menggunakan pewarnaan graf pada proses enkripsi dan dekripsinya. Metode enkripsi dan dekripsi pada protokol yang diusulkan mampu mencapai kecepatan enkripsi 512 byte data sebesar 0.0160 milidetik dan dekripsi 512 byte sebesar 0.0264 milidetik meningkat dari metode lain dengan proses enkripsinya 1.33 detik dan dekripsi 0.743 detik untuk data sebesar 338 byte[10]. Peluang maksimal IV_{client} terpecahkan adalah sebesar $3.719933267899012 \times 10^{41}$ pada IV dengan panjang 36 karakter.

DAFTAR PUSTAKA

[1] C. Braz dan J. M. Robert, "Security and Usability: The Case of the User Authentication Methods," dalam *18th International Conference of the Association Francophone d'Interaction Homme-Machine*, 2006, hal. 199-203.

[2] Authentication in an Internet Banking Environment, FFIEC Standard 2005, 2005.

[3] T. Ahmad, "Global and Local Feature-based Transformations for Fingerprint Data Protection," disertasi Ph.D, RMIT University, Melbourne, Australia, 2012.

[4] T. Ahmad dan S. Wang, "Generating Cancelable Biometric Template Using a Projection Line," dalam *ICARCV*, 2010, hal. 7-12.

[5] T. Ahmad, J. Hu, dan S. Wang, "Pair-polar coordinate based cancelable fingerprint template," *Pattern Recognition*, vol. 44, no. 10-11, hal. 2555-2564, Mar. 2011.

[6] N. Ratha, J. Connell, dan R. M. Bolle, "Cancelable Biometrics: A Case Study in Fingerprints," dalam *ICPR*, 2006, hal. 370-373.

[7] D. Maltoni, D. Maio, A. K. Jain, dan S. Prabhakar, *Handbook of Fingerprints*, edisi kedua, London: Springer, 2009.

[8] M. Kubale, *Graph Colorings*, Rhode Island: American Mathematical Society, 2004, hal. 95.

[9] W. Diffie dan M. Hellman, "New direction in cryptograph," *IEEE Transactions on Information Theory*, vol. 22, no. 6, hal. 644-654, 1976.

[10] K. Xi, T. Ahmad, F. Han, J. Hu, "A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environtment," dalam *Security and Communication Networks*, vol. 4, no. 5, hal. 487-499.

[11] K. Xi dan J. Hu, "Biometric Mobile Template Protection: A Composite Feature Based Fingerprint Fuzzy Fault," dalam *IEEE International Conference on Communication*, 2009, hal. 1-5.