PENGAMANAN DATA MENGGUNAKAN METODA ENKRIPSI SIMETRI DENGAN ALGORITMA FEAL

Semuil Tjiharjadi[1], Marvin Chandra Wijaya[2]
Jurusan Sistem Komputer, Fakultas Teknik, Universitas Kristen Maranatha
Jl. Suria Sumantri 65, Bandung 40164
Telp. (022) 2012186 ext. 228, Faks. (022) 2012186 ext. 230
E-mail: semuiltj@maranatha.edu; marvinchw@gmail.com

ABSTRAK

Dalam dunia sekarang ini, kemajuan teknologi di bidang komputer dan telekomunikasi berkembang sangat pesat. Lalu lintas pengiriman data dan informasi yang semakin global, serta konsep open system dari suatu jaringan memudahkan seseorang untuk masuk ke dalam jaringan tersebut. Hal tersebut dapat membuat proses pengiriman data menjadi tidak aman dan dapat saja dimanfaatkan oleh pihak lain yang tidak bertanggung jawab, yang mengambil informasi atau data yang dikirimkan tersebut di tengah perjalanan. Maka dibutuhkan suatu sistem keamanan yang dapat menjaga kerahasiaan suatu data, sehingga data tersebut dapat dikirimkan dengan aman. Salah satu solusi untuk menjaga keamanan dan kerahasiaan pada proses pengiriman data dengan menggunakan teknik kriptografi. Dalam makalah ini penulis merealisasikan suatu perangkat lunak enkripsi dan dekripsi file teks dengan menggunakan algoritma kriptografi yaitu FEAL. Di samping itu untuk meningkatkan keamanan pada perangkat lunak disertakan juga proses validasi dan proses digital signature, sehingga perangkat lunak dapat mendeteksi adanya perubahan data atau informasi yang dikirimkan serta menjamin keaslian pengirim informasi.

Kata Kunci: enkripsi, simetri, feal

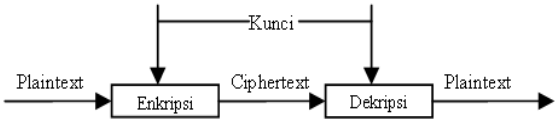
1. PENDAHULUAN

Masalah keamanan merupakan suatu aspek penting dalam pengiriman data maupun informasi melalui jaringan. Hal ini disebabkan karena kemajuan di bidang jaringan komputer dengan konsep *open system*-nya sehingga memudahkan seseorang untuk masuk ke dalam jaringan tersebut. Hal tersebut dapat mengakibatkan proses pengiriman data menjadi tidak aman dan dapat saja dimanfaatkan oleh orang maupun pihak lain yang tidak bertanggung jawab, untuk mengambil data ataupun informasi ditengah jalan. Oleh karena itu, dibutuhkan suatu sistem keamanan yang dapat menjaga kerahasiaan suatu data maupun informasi, sehingga data tersebut dapat dikirimkan dengan aman. Salah satu cara untuk menjaga keamanan dan kerahasiaan suatu data maupun informasi adalah dengan teknik enkripsi dan dekripsi guna membuat pesan, data, maupun informasi agar tidak dapat di baca atau di mengerti oleh sembarang orang, kecuali untuk penerima yang berhak.

2. ISI

Teknik pengamanan data menggunakan enkripsi dan dekripsi dikenal dengan nama kriptografi, sebagai sebuah ilmu atau seni untuk mengamankan pesan atau data dengan cara menyamarkan pesan tersebut sehingga hanya dapat dibaca oleh pengirim dan penerima pesan. Penerapan kriptografi pada komputer dapat menyamarkan pesan yang berupa file teks, gambar, suara, gambar bergerak dan lain-lain.

Secara global teknik enkripsi dan dekripsi data terdiri dari dua metoda, yaitu metoda kunci publik dan metoda kunci simetri. Metoda kunci simetri menggunakan password atau kata kunci yang sama untuk melakukan enkripsi dan juga dekripsi. Karena itu metoda ini sering juga disebut dengan metoda secret key Criptography. Contoh-contoh metoda ini adalah: DES (Data Encryption Standard), IDEA (International Data Encryption Algoritm), RC5, Blowfish, dan FEAL. Cara kerja metoda enkripsi ini terlihat seperti gambar 1.

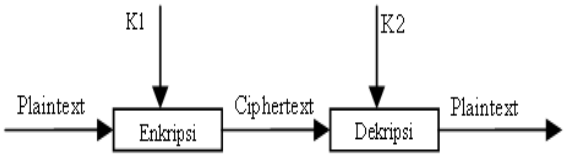


Gambar 1. Kriptografi kunci simetris

Masalah utama bagi metoda pengamanan data dengan kunci simetris adalah bagaimana mengirimkan kunci simetris tersebut dari pengirim kepada penerima. Tentunya akan metoda pengamanan ini tak akan berguna bila kunci sampai jatuh ke tangan orang yang tidak berhak.

Untuk itu dikembangkan metoda kunci asimetris yang dikenal juga kunci publik. Metoda kunci publik ini pertama kali ditemukan oleh Whitfield Diffie dan Martin Hellman, serta Ralph Merkle secara independent. RSA, El Gamal, dan Rabin adalah contoh-contoh kriptografi kunci simetris. Metoda ini menggunakan kunci yang berbeda untuk enkripsi dan dekripsi. Melalui metoda yang ada maka kunci enkripsi dapat dipublikasikan pada

pengirim tanpa khawatir jatuh ke tangan orang yang tidak berhak, karena hanya dapat digunakan untuk mengenkripsi dan tidak bisa digunakan untuk mengdekripsi. Kunci untuk mengenkripsi tersebut kemudian dikenal dengan nama kunci publik sedangkan kunci untuk mendekripsi dikenal dengan nama kunci privat. Cara kerjanya dapat dilihat pada gambar 2, K1 adalah kunci publik sedangkan K2 adalah kunci privat.



Gambar 2. Kriptografi kunci asimetris

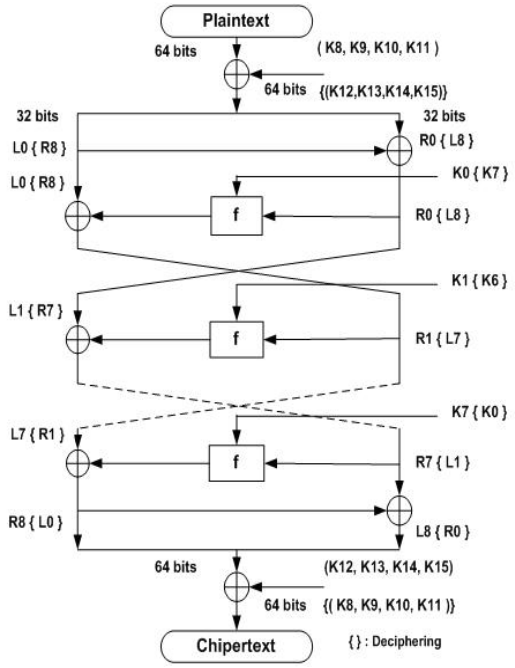
Pada makalah ini difokuskan pada penggunaan pengamanan data menggunakan kunci simetri dengan algoritma FEAL, namun makalah ini memiliki program untuk melakukan pendeteksian terhadap siapa pengirim menggunakan pengamanan tanda tangan digital yang menggunakan metoda algoritma RSA.

Algoritma-algoritma dalam proses enkripsi dan dekripsi sebenarnya terbagi menjadi dua macam, yaitu yang bersifat *stream cipher* dan *block cipher*. *Stream cipher* mengenkripsi *plaintext* atau mendekripsi *ciphertext* secara bit per bit, yaitu satu bit dienkripsi atau didekripsi per satuan waktu. Sedangkan algoritma tipe *block cipher* mengenkripsi *plaintext* dan mendekripsi *ciphertext* secara blok per blok (blok adalah kumpulan bit), yaitu satu blok dienkripsi atau didekripsi per satuan waktu. Tentu saja proses enkripsi dan dekripsi menggunakan metoda *block cipher* memiliki kecepatan yang lebih baik dibandingkan metoda *stream cipher*.

Untuk itu dipilih algoritma FEAL untuk makalah ini dikarenakan metoda ini memiliki kecepatan proses baik untuk enkripsi maupun dekripsi karena merupakan algoritma simetris yang memiliki algoritma yang bersifat *block cipher* sehingga jauh lebih cepat dibandingkan metoda enkripsi publik. Selain itu dirancang pula program untuk melakukan validasi dan tanda tangan digital untuk lebih menjamin proses pengamanan data secara keseluruhan.

2.1 Algoritma FEAL

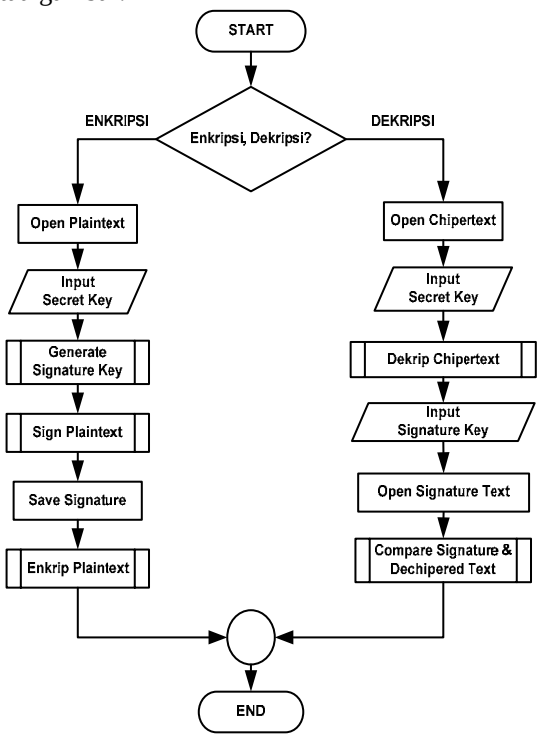
Akihiro Shimizu dan Shoji Miyaguchi dari NTT Jepang (1435). FEAL singkatan dari *Fast Encryption Algorithm*. FEAL merupakan enkipsi tipe simetris *block chippers*. FEAL mempunyai panjang blok 64 bit, panjang kunci 64 bit, dan memiliki iterasi sebanyak 8 ronde seperti terlihat pada gambar 3.



Gambar 3. Diagram Alir Algoritma FEAL

2.2 Perancangan Perangkat Lunak

Perangkat lunak enkripsi dan dekripsi LOKI-91 direalisasikan dengan menggunakan Microsoft Visual Basic, yang mendukung aplikasi GUI (*Graphical User Interface*) sehingga memungkinkan penggunanya berkomunikasi dengan modus grafik atau gambar.



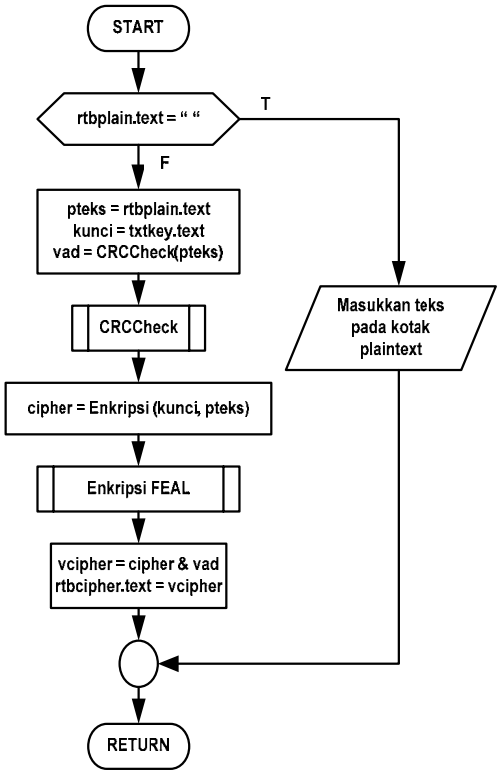
Gambar 4. Diagram Alir Program Pengamanan Data FEAL

Gambar 4 menampilkan flowchart program yang menampilkan 2 menu yang berisi menu untuk

enkripsi dan menú untuk dekripsi, juga terdapat kotak teks kunci serta tombol-tombol untuk menjalankan perintah tertentu. Pada gambar 4 ditampilkan perangkat lunak yang terdiri atas empat buah program utama, yaitu program enkripsi, program dekripsi, program validasi, dan program digital signatura, serta beberapa buah sub-program pendukung. Untuk melakukan enkripsi atau dekripsi, kotak teks masukkan dan kotak teks kunci harus diisi terlebih dahulu.

Untuk menú enkripsi, dalam tahap proses ini pengirim akan mengerjakan proses secara bertahap dari *open plaintext*, *input secret key*, sub rutin *generate signature key*, sub rutin *sign plaintext*, *save signature*, dan sub rutin enkrip *plaintext*.

Untuk menu dekripsi, dalam tahap proses ini penerima akan mengerjakan proses secara bertahap dari *open chipertext*, *input secret key*, sub rutin dekrip *chipertext*, input *signature key*, *open signature key*, dan *compare signature & dechipered text*.



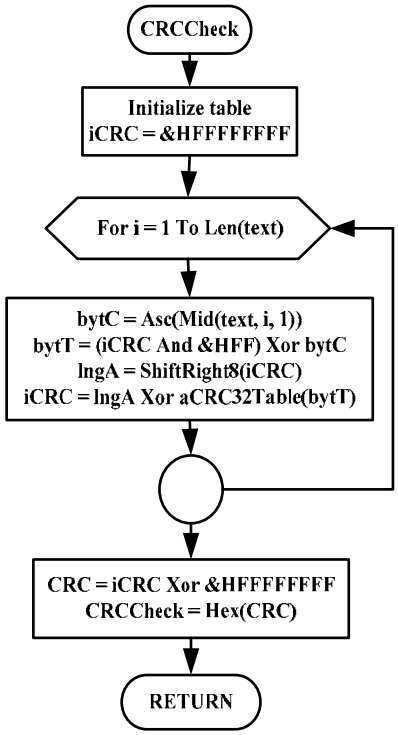
Gambar 5. Diagram Alir Eksekusi Enkripsi

2.3 Proses Enkripsi

Untuk proses enkripsi, pertama kali program akan memeriksa terlebih dahulu ada atau tidaknya teks pada kotak *plaintext* sebagai teks input proses enkripsi, jika teks belum dimasukkan maka akan ditampilkan pesan peringatan dan program akan dihentikan. Jika teks sudah dimasukkan maka akan dilakukan pengecekan karakter pada teks masukkan tersebut untuk validasi data. Pada fungsi enkrip, kemudian akan dicek jumlah karakter dari *plaintext*nya. Jika *plaintext* bukan kelipatan dari blok yang panjangnya 64 bit, maka akan ditambahkan angka "0" sebanyak jumlah karakter yang perlu ditambahkan agar jumlahnya merupakan kelipatan 64 bit dikurangi satu, dan karakter terakhir yang ditambahkan adalah nilai dari banyaknya karakter yang ditambahkan, seperti terlihat pada gambar 5.

2.4 Proses Validasi

Untuk proses validasi digunakan CRC cek. Pada CRC cek, pertama kali akan dilakukan inisialisasi tabel kemudian iCRC diatur sehingga semua bit-nya bernilai satu. Tabel CRC terdiri dari 256 buah nilai yang tiap nilainya mewakili satu karakter. Nilai ini dibuat secara acak, yang masing-masing memiliki panjang 64 bit. Setiap karakter pada teks diubah dalam kode ASCII kemudian di XOR dengan iCRC yang sebelumnya telah di AND dengan bilangan heksa FF, hasilnya akan menjadi indeks untuk table CRC. Keluaran dari table di XOR dengan iCRC yang telah digeser ke kanan sebanyak delapan bit dan hasilnya menjadi iCRC yang baru, kemudian karakter berikutnya akan diproses. Setelah semua karakter selesai diproses, iCRC di XOR dengan bilangan heksa FFFFFFFF menjadi CRC, terlihat pada gambar 6.



Gambar 6. Diagram Alir CRCCheck

2.5 Proses Digital Signature

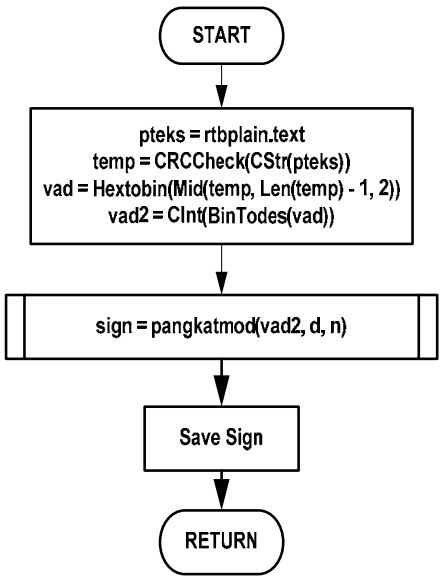
Proses *digital signature* ini menggunakan algoritma RSA dan input yang digunakan berupa nilai yang diambil dari nilai validasi. Program akan melakukan proses *Generate* untuk menghasilkan kunci publik dan kunci privat. Kunci privat akan digunakan untuk menandatangani *plaintext* dan

kemudian dihasilkan nilai *signature*. Dalam proses verifikasi, kunci publik yang telah dihasilkan akan digunakan beserta *signature* untuk memperoleh keaslian dari *plaintext*.

Algoritma dari proses pembangkit kunci tanda tangan dijital adalah sebagai berikut:

- Pilih dua buah bilangan prima yang berbeda (p dan q) (1)
- Hitung:
  $n = p * q$ (2)
- Hitung
  $\theta = (p - 1) * (q - 1)$ (3)
- Pilih sebuah integer e dengan batas $1 < e < \theta$ dan memenuhi syarat:
  $\gcd(e, \theta) = 1$ (4)
- Dengan menggunakan algoritma *extended Euclidean* dapat diperoleh nilai d, dengan batas $1 < d < \theta$ dan memenuhi syarat:
  $ed \equiv 1 \pmod{\theta}$ (5)
- Diperoleh kunci publik (e, n) dan kunci privat (d, n)
- Untuk memperoleh nilai *signature* digunakan persamaan:
  $s = H^d \bmod n$, (6)
  nilai H diambil dari nilai validasi
- Dan untuk verifikasi digunakan persamaan:
  $v = s^e \bmod n$ (7)

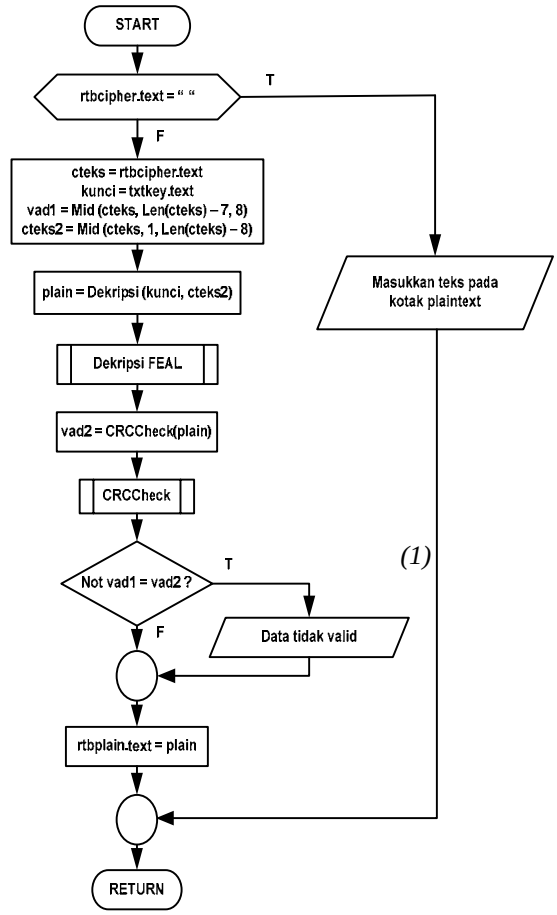
Untuk proses verifikasi nilai *signature* penerima harus mengetahui kunci publik dan nilai *signature*nya. Pada penerima kemudian akan dilakukan perhitungan verifikasi. Hasil dari perhitungan ini harus sesuai dengan nilai validasi yang digunakan oleh pengirim untuk menghasilkan nilai *signature*nya. Proses verifikasi akan dilakukan saat penerima mengeksekusi tombol *compare signature & dechipered text* seperti pada gambar 7.



Gambar 7. Diagram Alir Program Tanda Tangan Digital

2.6 Proses Dekripsi

Proses dekripsi algoritma FEAL sama dengan proses enkripsi, hanya pada proses dekripsi input yang digunakan berupa ciphertext dan proses kerjanya kebalikan dari proses enkripsi. Pada proses dekripsi urutan kunci yang digunakan merupakan kebalikan dari urutan kunci yang digunakan pada proses enkripsinya. Gambar 8 memperlihatkan proses dekripsi yang terjadi.



Gambar 8. Diagram Alir Eksekusi Dekripsi

3. PENGUJIAN

Pengujian perangkat lunak ini dilakukan pada komputer dengan CPU ber-processor Intel Pentium 4 1,7GHz, RAM 256 MB dengan sistem operasi Windows XP.

Berikut ini adalah beberapa pengujian yang dilakukan pada program enkripsi dan dekripsi dengan algoritma FEAL ini. Pengujian yang ditampilkan adalah pengujian terhadap proses enkripsi, proses dekripsi serta lamanya waktu yang diperlukan dalam melakukan proses enkripsi dan dekripsi tersebut. Data file yang digunakan adalah data berbentuk teks.

Plaintext yang digunakan :

"Kegagalan bukan merupakan akhir dari segalanya, melainkan sebuah awal dari proses menuju keberhasilan."

3.1 Proses Enkripsi (dilakukan pengirim) :

Kunci yang digunakan : elektroo

Nilai *Digital Signature*

Public Key : (2203,5959)

Private Key : (2667,5959)

NilaI *Signature :* 3784

Enkripsi

Ciphertext :

```
    ¹}ÒyG%—ºµ;™_
~;Vkú¾ D_áZ~Ï©eØÙqÐ__f_wx__+±_Ó$V\£û¹_jO
5ú ÿd¥k Ñ à.¶É!O ž¦#dˆ <›o–
```

Nilai Validasi

Kunci Validasi : E5833960

3.2 Proses Dekripsi (dilakukan penerima) :

Dekripsi

Kunci yang digunakan : elektroo

Nilai *Signature* yang dimasukkan :

Public Key : (2203, 5959)

Signature : 3784

Receive text :

> Kegagalan bukan merupakan akhir dari segalanya, melainkan sebuah awal dari proses menuju keberhasilan.

Jika pada penerima kunci dekripsi, nilai *signature* yang di inputkan berbeda dengan yang digunakan pada proses enkripsi serta bila terdapat perubahan pada ciphertext saat dikirimkan maka hasil yang diperoleh *(receive text)* tidak akan sesuai dengan *plaintext* yang dikirimkan.

3.3 Pengujian terhadap waktu proses

Berdasarkan beberapa proses pengujian dengan menggunakan beberapa file dengan ukuran yang berbeda-beda didapatkan waktu proses secara keseluruhan seperti pada tabel 1.

Tabel 1. Pengujian ukuran terhadap waktu proses

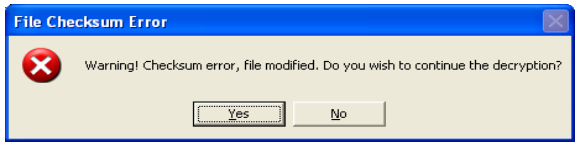
N0	Ukuran file	Waktu (detik)	
		Dekripsi	Enkripsi
1	10kb	1,5	1,6
2	20kb	3,2	3,3
3	40kb	6,5	6,6
4	60kb	10	11
5	90kb	15	15,5
6	120kb	20	21

3.4 Pengujian validasi menggunakan CRC Check

Untuk menguji apakah program mampu mendeteksi adanya bit yang berubah untuk proses validasi *cipher text*, maka *cipher text* sengaja diubah dengan menambahkan karakter ‘1’ sehingga *ciphertext* menjadi seperti berikut:

```
    1¹}ÒyG%—ºµ;™_
~;Vkú¾ D_áZ~Ï©eØÙqÐ__f_wx__+±_Ó$V\£û¹_jO
5ú ÿd¥k Ñ à.¶É!O ž¦#dˆ <›o–
```

mendeteksi bahwa file *cipher text* mengalami perubahan yang menghasilkan *checksum* yang berbeda dengan *checksum* yang disimpan pada file *ciphertext* sehingga muncul warning box berikut:



Gambar 9. Pesan kesalahan karena validasi gagal

3.5 Pengujian tanda tangan digital

Untuk menguji apakah program mampu mendeteksi potensi hacking maka program ditambahkan keamanannya dengan tanda tangan digital yang dapat mendeteksi apakah yang mengirimkan pesan adalah benar orang yang berhak. Maka pada pengujian ini ditampilkan bila tanda tangan digital yang seharusnya adalah:

Private Key : (2667,5959)

Diganti menjadi : (2668,5959), maka akan dihasilkan *ciphertext* sebagai berikut:

```
    7OïL#‘Íÿã8vøt} ã«¶äð–
º3• NÂ°Û ¨ îÙãt£ú›â¡gï â× …/{Ñ±.ˆ =lö
I]H,Š<ÛŠ“»kí«wý_¨º_Þ Ñîá3Ö
```

Maka saat program dijalankan maka program mendeteksi bahwa private key yang digunakan bukanlah pasangan dari public key, sehingga dapat diketahui bahwa terjadi perbedaan pada tanda tangan digital.

4. KESIMPULAN

Berdasarkan hasil pengamatan dan pengujian yang telah dilakukan maka dapat disimpulkan bahwa dalam makalah ini :

1. Perangkat lunak enkripsi dan dekripsi data dengan metoda kunci simetri menggunakan algoritma FEAL telah berhasil direalisasikan dan berjalan dengan baik.
2. Proses Validasi dan *Digital Signature*, yang ditambahkan pada perangkat lunak dapat berjalan dengan baik dan dapat mendeteksi adanya perubahan pada teks yang dikirim, serta dapat menjamin keutuhan dan keaslian data dari pengirim informasi.
3. Waktu untuk proses enkripsi dan dekripsi berbanding lurus dengan penambahan ukuran file. Nilai rata-rata kecepatan proses yang dihasilkan sebesar 6 KB/detik.

PUSTAKA

A.Menezes, P.van Oorschot, and S.Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.

en.wikibooks.org

Kahate, Atul, *Cryptography and Network Security*, Tata McGraw-Hill, 2003.

Rosen, Kenneth H, *Discrete Matehematics and Its Applications, 5th edition*, McGraw-Hill, 2003.

Schneier, Bruce; "*Applied Cryptography Second Edition: protocol, algorithm, and source code in C*"; John Wiley and Son , 1996.

Stallings, William, *Network and Internetwork Security : principles and practise*, Prentice Hall, New Jersey, 1995.