Pengamanan Data Menggunakan Kriptografi EC *Signcryption* dan Steganografi *Least Significant Bit* (LSB)

***Dini Triasanti*[1a]**
[1]*Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma*
Jl. Margonda Raya No. 100, Pondok Cina, Depok 16424

[a]*dini3asa@staff.gunadarma.ac.id*

Abstraksi

Penelitian ini bertujuan untuk mengembangkan sistem pengamanan ganda terhadap data yaitu signcryption *(*signature-encryption*) data penyembunyian data terenkripsi tersebut dalam media teks, atau gambar, atau audio, atau video. Dalam tulisan ini metode kriptografi yang digunakan adalah* Elliptic Curve Signcryption *sedangkan steganografi yang digunakan adalah metode Modifikasi LSB (*Least Significant Bit*). Sistem ini berhasil dikembangkan dengan menggunakan bahasa pemrograman Java, operasi* embedding *dan* retrieving *dapat dilakukan dengan cepat, tidak bergantung secara signifikan terhadap perubahan ukuran file stego, dari segi HVS, HAS, dan kombinasinya tidak ada perubahan kualitas gambar dan suara pada file stego, fitur kompresi, enkripsi dan signature (signcrypt), serta steganografi dapat memberikan pengamanan data yang berlapis.*

***Kata Kunci**: Kriptografi, Steganografi, Elliptic Curve Signcryption, Least Significant Bit, Java.*

Data Protection Using EC Signcryption Cryptograph and Least Significant Bit (LSB) Steganography

Abstract

The study aims at developing multiple security system of data namely data signcryption (signature-encryption). The concealment of the encryptioned data in text, or image, or audio, or video. In the study, the cryptograph method used is Elliptic Curve Signcryption, while steganograph used is modificated LSB (Least Significant Bit) method. The system is succeed to be developed using Java program, embedding and retrieving operation can be done fast, it is not significantly depended on the change of stego file. From the side of HVS, HAS, and the combination, there is no change of picture and audio quality in stego file, compress fiture, encryption and signature (signcrypt), as well as the steganograph can provide multi-protection of data.

***Keywords**: Cryptograph, Steganograph, Elliptic Curve Signcryption, Least Significant Bit, Java.*

PENDAHULUAN

Kriptografi berkaitan dengan enkripsi data dari sisi pengirim dan dekripsi data dari sisi penerima. Aspek keamanan informasi yang merupakan tujuan kriptografi menyediakan kerahasiaan data, otentikasi, integritas data dan nonrepudiasi [Munir, 2006]. Saat ini pendekatan standar untuk mencapai kerahasiaan dan keaslian pesan adalah *signature* diikuti enkripsi, yaitu pengirim akan menandatangani pesan menggunakan skema tandatangan digital, kemudian mengenkripsi pesan yang telah ditanda-tangani dengan menggunakan algoritma enkripsi. Skema digital *signcryption* merupakan salah satu metode kriptografi yang memenuhi dua fungsi, yaitu enkripsi secara aman (*secure encryption*) dan tandatangan digital (*digital signature*) secara bersamaan, sehingga membutuhkan *cost* yang lebih kecil daripada menggunakan skema *signature* kemudian *encryption* [Zheng, 1998].

Steganografi adalah ilmu yang mempelajari cara penyembunyian informasi pada suatu media sehingga keberadaannya tidak ter-deteksi oleh pihak lain yang tidak berwenang atas informasi tersebut. Steganografi dapat bermanfaat untuk melindungi informasi yang terdapat di dalam file digital yang berupa file gambar, audio, atau video. Kombinasi kriptografi dan steganografi diimplementasikan dengan tujuan agar diperoleh tingkat pengamanan yang berlapis terhadap data rahasia. Dalam penelitian ini mula-mula dilakukan *signature* dan enkripsi (*signcrypt*) data menggunakan skema *Elliptic Curve Signcryption*. Selanjutnya proses stega-nografi dilakukan dengan menyisipkan data ter-*signcrypt* ke dalam file tertentu dengan teknik modifikasi LSB (*Least Significant Bit*).

Tujuan dari penelitian ini adalah mengembangkan sistem keamanan data yang menerapkan ide *signcryption* dan steganografi yaitu mengamankan data rahasia dengan kriptografi cepat untuk men-*signcrypt* dan mengaman-kan data tersebut di dalam file digital seperti file gambar, audio, atau video, sehingga akan diperoleh pengamanan berlapis terhadap data rahasia. Sistem ini dikembangkan menggunakan JDK 1.7.0_03.

TINJAUAN PUSTAKA

Steganografi adalah teknik yang digunakan untuk menyembunyikan data rahasia menggunakan media file tertentu. Jenis data rahasia dapat berupa pesan atau file digital dan jenis media file dapat berupa berbagai jenis file digital [Triasanti, 2008]. Cara paling umum untuk menyembunyikan data rahasia adalah dengan memanfaatkan *Least Significant Bit* (LSB). Menurut [Sridevi 2005] *Enhanced Audio Steganography* (EAS) adalah satu sistem yang diusulkan berdasarkan pada audio Steganografi dan kriptografi, memastikan keamanan pengiriman data antara sumber dan tujuan. EAS menggu-nakan algoritma enkripsi yang sangat *powerfull* pada keamanan level pertama yang sangat kompleks. Pada keamanan level kedua digunakan algoritma modifikasi LSB (*Least Significant Bit*) yang *powerfull* untuk encode pesan ke dalam audio.

Terdapat banyak algoritma enkripsi data yang diusulkan untuk menyembunyikan data dari penyusup tetapi hampir semua algoritma yang diusulkan memiliki beberapa kelemahan. Pada penelitian ini digunakan *signcryption* dan steganografi [Thorat, 2012]. Dalam [Zheng, 1996] mengenai pertanyaan tentang *cost* pengiriman/ penyimpanan pesan yang aman dan otentik, dapat dilihat dari persamaan: *Cost (Signature & Encryption) << Cost (Signature) + Cost(Encryption)*, Secara khusus, Zheng menemukan kriptografi primitive baru disebut sebagai *signcryption* yang sekaligus memenuhi kedua fungsi tanda tangan digital dan enkripsi kunci publik dalam sebuah langkah logis [Zheng, 1996]. Dari persamaan tersebut, skema *signcryption (Signature & Encryption)* menghasilkan *ciphertext* yang lebih pendek, lebih efisien secara komputasi,dan lebih besar fungsi jaminan keamanan daripada

melakukan kombinasi tanda tangan digital dan enkripsi [Dini, 2012].

Skema dari *EC Signcryption* membutuhkan algoritma sebagai berikut [Mohamed, 2009]:

Parameter publik

C: Kurva Elliptic atas GF(p^h) dengan $p \geq 2^{150}$ dan gh = 1.

q: sebuah bilangan prima besar yang ukurannya sekitar $|p^h|$.

G: titik dengan order q, yang dipilih secara acak dari titik-titik pada C.

hash: fungsi hash satu arah.

$hash_k$: fungsi hash satu arah berkunci.

E, D: algoritma enkripsi dan dekripsi dari kunci privat cipher.

Kunci Alice

v_a: kunci privat Alice, yang dipilih seragam secara acak dari [1, … q - 1].

P_a: kunci publik Alice, $P_a = v_a G$ sebuah titik pada C.

Kunci Bob

v_b: kunci privat Bob, yang dipilih seragam secara acak dari [1, … q - 1].

P_b: kunci publik Bob, $P_b = v_b G$ sebuah titik pada C.

Signcryption message oleh pengirim Alice

v ε R[1, …, q - 1] bilangan acak.
$k_1 = hash(vG)$
$k_2 = hash(v P_b)$
$c = E\ k_2\ (m)$
$r = hash(c, k_1)$
$s = v/(r + v_a) \bmod q$
$R = rG$
Kirim c, R, s kepada Bob

Unsigncryption c, R, s oleh penerima Bob

$k_1 = hash(s(R + P_a))$
$r = hash_{k1}(c)$
$k_2 = hash(v_b s(R + P_a))$
$m = D_{k2}(c)$
c memiliki signature yang valid/hanya berlaku jika rG = R

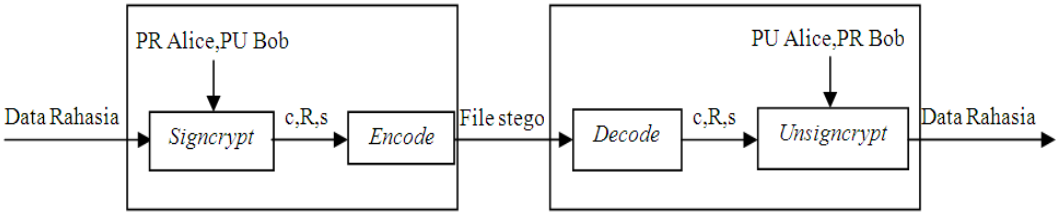
Verifikasi c, R, s oleh third party/pihak ketiga tanpa penyingkapan dari m

$k_1 = hash(s(R + P_a))$
$r = hash_{k1}(c)$
c memiliki signature yang valid/hanya berlaku jika rG = R

JDK 1.7.0_03 adalah Java Development Kit versi 1.7.0_03, paket pengembangan Java untuk kompilasi dan eksekusi program Java yang didalamnya terdapat *Java Runtime Envi-ronment* dan *Java Virtual Machine*. Java 2 Standard Edition (J2SE) menyediakan lingkungan pengembangan yang memiliki banyak fitur, stabil, aman, dan dapat berjalan pada berbagai sistem operasi. J2SE mampu mendukung rancangan antar muka, masukan/keluaran, dan pemrograman jaringan serta menyediakan paket-paket dalam library Java [Kadir, 2004].

METODE PENELITIAN

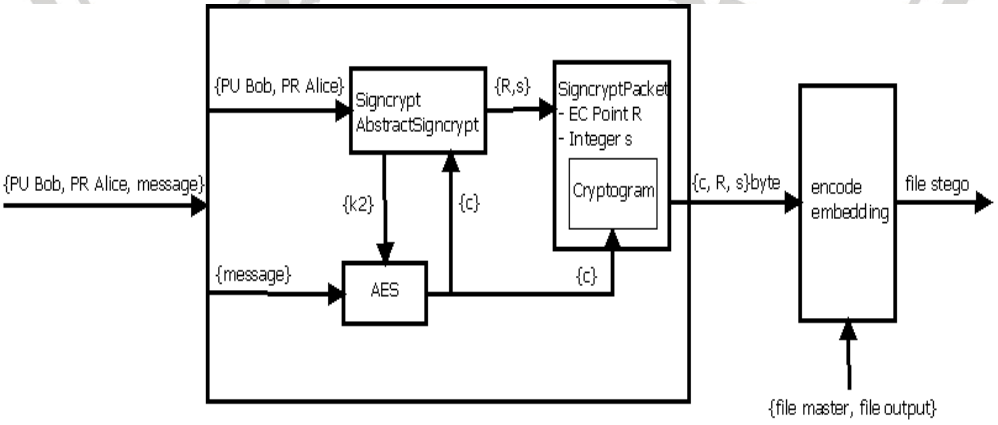
Jalan penelitian sistem keamanan data rahasia menggunakan kombinasi Kriptografi metode *EC Signcryption* dan Steganografi metode Modifikasi LSB (*Least Significant Bit*) dapat dilihat pada Gambar 1., dijelaskan bahwa masukkan dalam sistem adalah data rahasia dapat berupa pesan atau file digital, *password* yang akan membangkitkan kunci privat dan kunci publik, masukkan tersebut melewati proses enkripsi dan *signature* (*signcrypt*) menggunakan metode *EC Signcryption* menghasilkan nilai c, R, s. Selanjutnya nilai tersebut dikirim-kan ke dalam proses *encode* untuk disisipkan ke media file tertentu, proses tersebut menghasilkan file stego. Untuk mendapatkan keluaran data rahasia, file stego melewati proses *decode* menghasilkan nilai c, R, s yang selanjutnya me-lewati proses *unsigncrypt* yang membutuhkan masukkan *password* untuk membangkitkan kunci privat dan kunci publik.



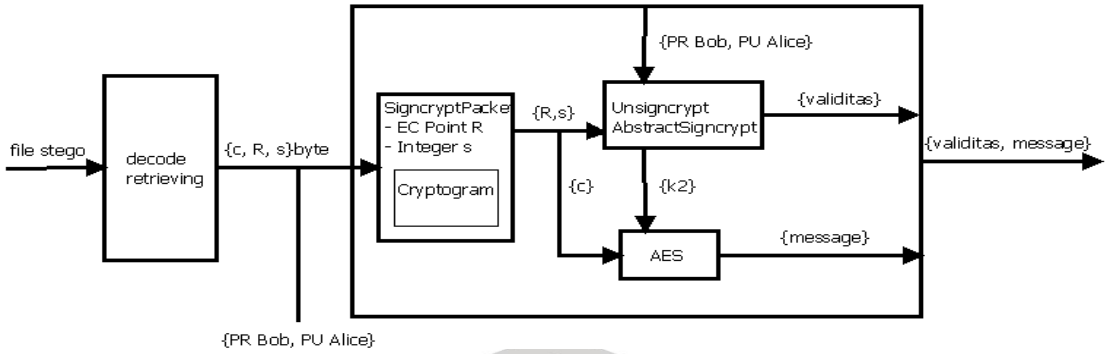
Gambar 1. Alur Proses Pengamanan Data Rahasia menggunakan Kombinasi *EC Signcryption* dan Steganografi

Pada Gambar 2, proses men-*signcrypt* pesan ditampilkan. Model proses meng-adopsi dari [Mehmet, 2011] digabungkan dengan konsep steganografi. Untuk *signcrypt* pesan, masukan pada skema *signcryption* adalah kunci publik penerima, kunci privat pengirim dan pesan. Kunci ini yang pertama kali digunakan di kelas signcrypt untuk membangkitkan kunci enkripsi simetrik K2. Kelas AES melakukan enkripsi pesan dengan K2 dan mengirimkan kembali ke kelas Signcrypt untuk membang-kitkan *EC point* R dan integer s. Seperti data yang dikirim, dalam banyak kasus ditangani sebagai byte, kelas SignCryptPacket, membungkus c, R, s ke dalam paket byte. Penulis sekarang memiliki pesan yang telah dienkripsi dan ditandatangani (*signcrypt*) dalam paket byte yang kemudian akan dikodekan/*encode* melalui proses *embedding* ke media file tertentu/objek stego menggunakan metode Modifikasi LSB (*Least Significant Bit*).

Setelah penulis menunjukkan proses *signcrypt*, pada gambar 3. akan ditunjukkan pro-ses *unsigncrypt* pesan yang mengadopsi dari [Mehmet, 2011] digabungkan dengan konsep steganografi. Mula-mula file stego didekodekan/*retrieving* menggunakan meto-de Modifikasi *Least Significant Bit* (LSB), menghasilkan nilai c, R, s dalam byte. *Unsigncryption* digunakan untuk mem-validasi dan mendekripsi pesan. Kemudian kelas SignCryptPacket yang fungsinya membungkus komponen c, R, dan s pada proses *signcrypt*, akan mengeluarkan komponen tersebut pada proses *unsigncrypt*. Dengan kunci publik dari pengirim, kunci privat dari penerima, AES ciphertext c, *EC Point* R dan integer s, kelas Unsigncrypt dapat menghitung AES sime-trik kunci K2 dan memastikan pesan tersebut memiliki tanda tangan yang valid atau tidak. Perhatikan bahwa validitas dapat diverifikasi tanpa mendekripsi ciphertext c. Kelas AES mendekripsi ciphertext c dengan *session key* K2, dan penulis mendapatkan satu komponen, yaitu pesan m. Hasil akhir berupa *message* dan validitas dapat dilihat pada Gambar 3.



Gambar 2. Proses *Signcrypt*

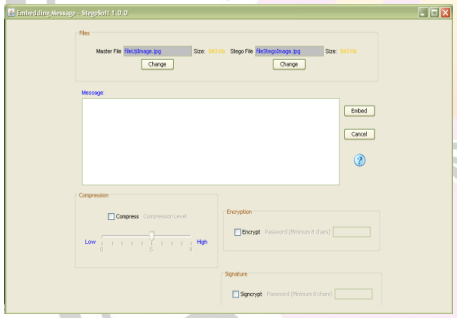


Gambar 3. Proses *Unsigncrypt*

PEMBAHASAN

Uji coba Embed Message

Uji coba menyisipkan pesan rahasia akan menghasilkan satu file stego. Lihat Gambar 4. memilih file *master* dan file stego. Di dalam area teks pesan, pengguna dapat menulis pesan yang akan disembunyikan. Proses selanjutnya akan digunakan fitur enkripsi dan *signature* (*signcrypt*) menggunakan 2 *password* tertentu. Operasi kompresi dan *signcrypt* akan sangat berguna untuk menjaga file stego tidak diketahui oleh pihak yang tidak berwenang. Setelah memilih tombol *embed*, akan menampilkan Gambar 5. *message box* yang menandai suksesnya operasi tersebut.

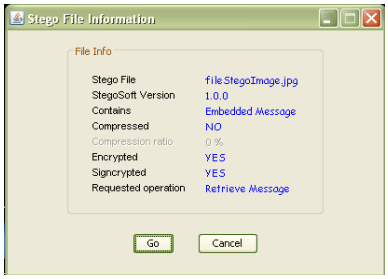


Gambar 4. Form *Embed Message*

Gambar 5. *Message Box* Operasi *Embed Message* Berhasil

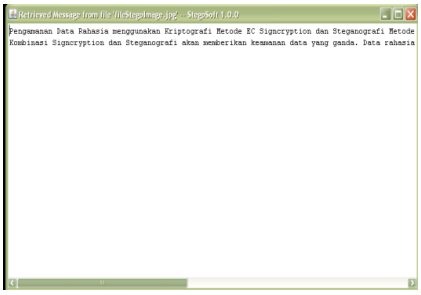
Uji Coba Retrieve Message

Uji coba ini untuk mendapatkan kembali pesan rahasia yang disembunyikan, pengguna harus memilih file stego. Berikutnya, Gambar 6. form *file information* akan muncul yang berisi informasi bagi pengguna untuk me-*retrieve* pesan. Dalam kaitan dengan penggunaan fasilitas enkripsi dan *signature* (*signcrypt*) di dalam proses penyisipan pesan, setelah pemilihan tombol *Go* di dalam form *file information*, Gambar 7. *message box password* akan muncul. Setelah mengisi *password*, pesan rahasia akan muncul pada jendela yang berbeda seperti Gambar 8.



Gambar 6. Form *File Information Retrieve Message*

Gambar 7. *Message Box Password*



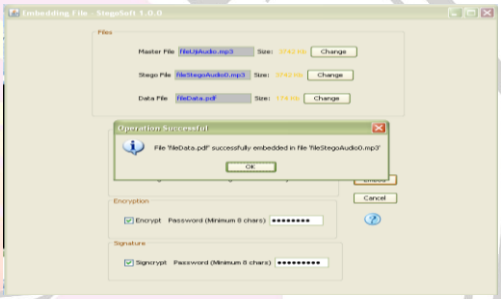
Gambar 8. Pesan di-*retrieve*

Uji Coba Embed File

Uji coba menyisipkan file yang akan disembunyikan akan menghasilkan satu file stego. Lihat Gambar 9. memilih file *master*, file stego dan file data. Proses selanjutnya menggunakan fitur enkripsi dan *signature* (*signcrypt*) menggunakan 2 password tertentu. Setelah memilih tombol *embed*, akan muncul Gambar10.

.

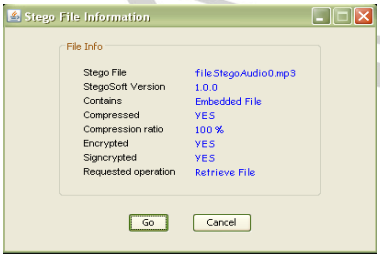
Gambar 9. Form *Embed File*



Gambar 10. *Message Box* Operasi *Embed File* Berhasil

Uji Coba Retrieve File

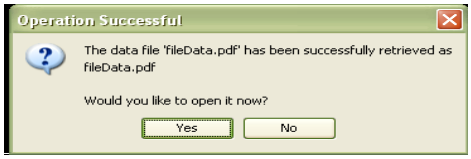
Pengguna harus memilih file stego. Berikutnya, Gambar 11. yang menampilkan informasi diperlukan bagi pengguna untuk mendapatkan kembali file yang tersembunyi akan muncul. Setelah pemilihan tombol *go*, Gambar 12. *message box password* akan muncul. Setelah pengguna mengisi *password*, akan menampilkan Gambar 13. *message box* yang menandai suksesnya operasi tersebut. Ketika pengguna memilih tombol *yes*, file yang tersembunyi akan muncul pada jendela *internet explorer* ditunjukkan pada Gambar 14.



Gambar 11. Form *File Information Retrieve File*

Gambar 14. File di-*retrieve*

Gambar 12. *Message Box Password*



Gambar 13. *Message Box* Operasi *Retrieve File* Berhasil

Statistik Uji Coba

Sistem ini telah melalui suatu rangkaian uji coba *embed* dan *retrieve* pesan atau file. Hasil pengujian dari segi waktu pengerjaan operasi *embed* dan *retrieve* pesan / file relatif singkat.

Tabel 1. Ringkasan Hasil Uji Coba berdasarkan ukuran file stego (tanpa kompresi)

Data rahasia yang disisipkan			**Ukuran file Stego**		
			Gambar	**Audio**	**Video**
Kategori	**Nama**	**Ukuran**	**.jpg** 843 Kb	**.mp3** 3,65 Mb	**.avi** 26,7 Mb
Pesan		340 Karakter	843 Kb	3,65 Mb	26,7 Mb
Teks	fileData.txt	6,58 Kb	845 Kb	3,65 Mb	26,7 Mb
	fileData.rtf	30,2 Kb	847 Kb	3,65 Mb	26,7 Mb
	fileData.doc	137 Kb	952 Kb	3,76 Mb	26,8 Mb
	fileData.pdf	175 Kb	1003,52Kb	3,81 Mb	26,9 Mb
	fileData.xls	67,2 Kb	896 Kb	3,70 Mb	26,7 Mb
	fileData.ppt	123 Kb	888 Kb	3,69 Mb	26,7 Mb
Image	fileData.jpg	18 Kb	860 Kb	3,67 Mb	26,7 Mb
	fileData.tif	19,6 Kb	855 Kb	3,66 Mb	26,7 Mb
	fileData.gif	9,15 Kb	852 Kb	3,66 Mb	26,7 Mb
	fileData.bmp	120 Kb	886 Kb	3,69 Mb	26,7 Mb

Hasil pengujian *embed message* dan *file* terhadap file *master* berupa file gambar, audio dan video disisipkan file data dari berbagai tipe yang terlihat pada tabel 1. membuktikan bahwa perubahan ukuran file tidak terlalu signifikan.

Tabel 2. Ringkasan Hasil Uji Coba *Embed Message* dengan panjang pesan yang sama berdasarkan fitur level kompresi mempengaruhi ukuran file stego

Level Kompresi	**Ukuran File**		
	File Gambar (fileUjiImage.jpg) 863.282 byte	**File Audio (fileUjiAudio.mp3)** 3.831.882 byte	**File Video (fileUjiVideo.avi)** 28.041.216 byte
0	863.771 byte	3.832.371 byte	28.041.705 byte
1	863.627 byte	3.832.227 byte	28.041.561 byte
2	863.627 byte	3.832.227 byte	28.041.561 byte
3	863.627 byte	3.832.227 byte	28.041.561 byte
4	863.627 byte	3.832.227 byte	28.041.561 byte
5	863.627 byte	3.832.227 byte	28.041.561 byte
6	863.627 byte	3.832.227 byte	28.041.561 byte
7	863.627 byte	3.832.227 byte	28.041.561 byte
8	863.627 byte	3.832.227 byte	28.041.561 byte
9	863.627 byte	3.832.227 byte	28.041.561 byte

Pengujian menggunakan fitur kompresi, ukuran file stego dapat dikurangi menjadi ukuran maksimum tujuannya untuk menghapuskan kecurigaan dari pihak yang tidak berwenang. Pada level kompresi berbeda yang diterapkan, masih tetap tidak ada

perbedaan yang berarti yang di-hubungkan dengan aspek indera pendengaran dan penglihatan pada setiap file stego. Kompresi ha-nya mempengaruhi ukuran file dan durasi waktu file (pada pengujian file audio dalam proses *embed*-file).

Tabel 3. Ringkasan Hasil Uji Coba *Embed File* dengan file yang sama berdasarkan fitur level kompresi mempengaruhi ukuran file stego

Level Kompresi	Ukuran File		
	File Gambar (fileUjiImage.jpg) 863.282 byte	**File Audio (fileUjiAudio.mp3)** 3.831.882 byte	**File Video (fileUjiVideo.avi)** 28.041.216 byte
0	1.041.683 byte	4.010.283 byte	28.219.617 byte
1	1.036.179 byte	4.004.779 byte	28.214.113 byte
2	1.036.019 byte	4.004.619 byte	28.213.953 byte
3	1.035.939 byte	4.004.539 byte	28.213.873 byte
4	1.036.643 byte	4.005.243 byte	28.214.577 byte
5	1.036.603 byte	4.005.203 byte	28.214.537 byte
6	1.036.603 byte	4.005.203 byte	28.214.537 byte
7	1.036.603 byte	4.005.203 byte	28.214.537 byte
8	1.036.603 byte	4.005.203 byte	28.214.537 byte
9	1.036.603 byte	4.005.203 byte	28.214.537 byte

Tabel 4. Perbedaan Durasi Waktu antara File *Master* dan File Stego

Nama File	Status	Durasi	Perbedaan durasi
fileUjiAudio.mp3	file *master*	3 menit 59 detik	-
fileStegoAudio0.mp3 – fileStegoAudio9.mp3	file stego Kompresi = 0-9	4 menit 10 detik	11 detik

Pengujian terhadap file stego yang dihasilkan dari proses *embed message* dan *file*, tidak ada perbedaan yang berarti pada visual dan audio yang dibandingkan dengan file *master*. Pengujian dilakukan meng-gunakan *Human Visual System* (HVS) untuk file gambar, *Human Auditory System* (HAS) untuk file audio, dan kombinasi dari HVS dan HAS untuk mengamati file video.

SIMPULAN DAN SARAN

Simpulan

Berdasarkan hasil uji coba dan pembahasan mengenai pengamanan data rahasia menggunakan kriptografi metode *EC Signcryption* dan Steganografi metode Modifikasi LSB (*Least Significant Bit*) dapat diambil kesimpulan yang berhubungan dengan pencapaian dari sistem ini adalah :

1. Java dapat mengembangkan sistem ini, karena Java merupakan pemrograman berorientasi objek memiliki paket *library* yang sangat luas, dengan mudah mengubah, menambah dan menggu-nakan kelas yang telah dibuat ke dalam sistem.
2. Sistem ini memenuhi persyaratan guna menyisipkan (*embed*) data rahasia berupa pesan maupun file data, tanpa merusak file *master* ataupun data yang telah disisipkan.
3. Sistem ini juga memenuhi persyaratan guna mendapatkan kembali (*retrieve*) data rahasia berupa pesan maupun file data pada file stego, tanpa diketahui oleh pihak yang tidak berwenang.
4. Sistem ini mendukung berbagai tipe file sehingga pengguna dapat memiliki pilihan yang lebih baik di dalam proses *embed* dan *retrieve* pesan atau file.
5. Sistem ini menawarkan fitur pengama-nan data berlapis yaitu dengan melaku-kan *sign-crypt* (enkripsi dan *signature*) terlebih dahulu sebelum disisipkan ke dalam file *output* (steganografi).

6. Sistem ini didukung oleh fitur kompresi, dengan menggunakan fitur ini ukuran file data dapat dikurangi menjadi ukuran yang maksimum dengan tujuan untuk menghapuskan kecurigaan dari pihak yang tidak berwenang.
7. Hasil uji coba operasi *embed* dan *retrieve* pesan/ ile, waktu pengerjaan setiap operasi relatif singkat dan sistem ini memberikan tampilan yang mudah dioperasikan oleh pengguna.
8. Hasil uji coba terhadap file *master* berupa file gambar, audio, atau video, perubahan ukuran file tidak terlalu signifikan pada proses *embed message* dan *file*.
9. Hasil uji coba penggunaan fitur kompresi hanya akan mempengaruhi ukuran file dan durasi waktu file, jika pengujian dilakukan pada file audio dalam proses *embed* file).
10. Hasil uji coba file stego, tidak ada perbedaan yang berarti pada HVS untuk file gambar, HAS untuk file audio, atau kombinasi HVS dan HAS untuk mengamati video.

Saran

Dari analisis yang dilakukan terhadap penelitian ini ada beberapa hal yang harus di-perhatikan antara lain file *master* yang bertindak sebagai media yang dapat disisipi sebaiknya dipilih dengan kapasitas yang cukup memadai untuk menampung file data rahasia, Pengembangan terhadap sistem agar mendukung perlindungan terhadap infor-masi seperti kunci sistem dan menyaji-kan nilai c, R, s yang dihasilkan dari metode *EC Signcryption* dalam tampilan sistem.

DAFTAR PUSTAKA

[Dini, 2012] Dini Handayani, Annisa, et all. Diakses tahun 2012. *Practical Signcryption untuk Transmisi Data Hasil Pemilu*. Lembaga Sandi Negara. Jakarta.

[Kadir, 2004] Abdul Kadir. 2004. *Dasar Pemrograman Java 2*. Andi Offset. Yogyakarta. [Mehmet, 2011] Mehmet, Jo. 2011. *End To End Data Protection of SMS Messages*. *Department of Telematics*. Norwe-gian University of Science Technology. Norwegia.

[Mohamed, 2009] Mohamed, Elsayed, Hasim Elkamchouchi. 2009. *Elliptic Curve Signcryption with Encrypted Message Autentication and Forward Secrecy*. [IJCSNS] International Journal of Computer Science and Network Security, VOL.9 No.1.

[Munir, 2006] Rinaldi Munir. 2006. *Kriptografi*, Informatika, Bandung.

[Sridevi, 2005] Sridevi, R, et all. 2005-2009. *Efficient Method of Audio Steganography by Modified Lsb Algorithm and Strong Encry-ption Key with Enhanced Security*. JATIT. JNTUCEH, Hyderabad. India. [Zheng, 1998] Zheng, Yuliang, Hideki Imai. 1998. *How to construct efficient signcryption schemes on elliptic curves*. Monash University, Australia and University of Tokyo, Japan.

[Thorat, 2012] Thorat, Suryakant, Madhav Bokare. 2012. *A Dynamic Method to Secure Confidential Data Using Signcryption with Steganography*. [IJESAT] International Journal Of Engineering Science & Advanced Technology, Volume-2, Issue-2, 183 – 191.

[Triasanti, 2008] Dini Triasanti. 2008. *Implementasi Perangkat Lunak Steganografi menggunakan Bahasa Pemrograman Java*. Skripsi Teknik Informatika. Universitas Gunadarma. Jakarta.

[Zheng, 1996] Zheng, Yuliang. 1996. *Digital Signcryption or How to Achieve Cost(Signature & Encryption) << Cost(Signature) + Cost (Encryption)*. Monash University. Australia.