Penandaan Citra dengan menggunakan Elliptic Curve Digital Signature Algorithm

Jordan Fernando / 13510069[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*fernandojordan.92@gmail.com*

***Abstract*—Citra merupakan salah satu media informasi yang sering digunakan. Citra banyak digunakan untuk merepresentasikan kondisi dari lingkungan, karya seni, dan bukti-bukti nyata. Bahkan di bidang penelitian sering menggunakan citra untuk mendukung kegiatan penelitian, seperti foto-foto yang diambil oleh robot NASA pada planet Mars. Saat ini citra dapat dibentuk dengan mudah karena kemajuan teknologi dan adanya piranti-piranti yang mendukung pembuatan citra. Oleh karena hal tersebut, citra juga rentan terhadap pemalsuan. Dalam makalah ini, saya akan membahas tentang cara penandaan citra dengan Elliptic Curve Digital Signature Algorithm. Bagian pertama akan membicarakan tentang apa itu citra beserta penggunaannya. Bagian kedua akan membicarakan tentang algoritma kriptografi berbasis Elliptic Curve, Digital Signature Algorithm, dan Elliptic Curve Digital Signature Algorithm. Bagian ketiga akan membicarakan tentang hasil implementasi dari Elliptic Curve Digital Signature Algorithm disertai dengan contoh-contoh. Bagian keempat akan membahas tentang perbandingan algoritma yang menggunakan Elliptic Curve dengan algoritma biasa. Terakhir, bagian kelima merupakan kesimpulan.**

***Index Terms*—citra, digital signature, elliptic curve.**

I. PENDAHULUAN

Citra merupakan kombinasi dari titik, garis, bidang, dan warna yang membentuk imitasi dari objek-objek fisik yang ada. Citra dapat berwujud dua dimensi ataupun tiga dimensi. Namun, citra yang dibahas dalam makalah ini adalah citra dua dimensi khususnya citra digital.

Citra memiliki banyak fungsi karena dapat memvisualisasikan informasi yang ada. Contoh fungsi-fungsinya antaralain sebagai karya seni, sebagai bukti, sebagai pembantu penelitian, sebagai hiburan, dan sebagai media informasi. Banyak informasi-informasi penting yang disebarkan menggunakan citra seperti foto-foto planet Mars yang diambil oleh robot NASA, foto-foto tingkat mikro untuk melihat bentuk dan sifat dari mikroba, dan foto-foto yang diambil untuk memberitahukan keadaan dari suatu lingkungan.

Citra digital tersusun oleh angka-angka biner yang diatur dalam format tertentu. Format-format yang paling umum digunakan antara lain JPEG, GIF, BMP, dan PNG. Format JPEG (Joint Photographic Experts Group) adalah format yang terkompresi. Format JPEG mendukung adanya 8-bit warna *grayscale* dan 24-bit warna biasa yang terdiri dari merah, hijau, dan biru. Format JPEG biasanya mengalami degradasi kualitas ketika disalin berkali-kali. Format GIF (Graphics Interchange Format) juga merupakan format terkompresi namun hanya dapat menampilkan 256 jenis warna. Kompresi pada GIF akan efektif jika gambar tidak bersifat detail. Format GIF juga mendukung adanya animasi dan masih sering digunakan. BMP (Bitmap) merupakan format yang digunakan oleh sistem operasi Windows. Format BMP tidak terkompresi dan sangat simpel dan mudah diproses. Format PNG (Portable Network Graphics) merupakan format yang bersifat *open source* sebagai penerus dari format GIF. Format PNG mendukung 8 bit warna gambar yang didukung derngan opsi transparansi dan 24 bit atau 48 bit warna sebenarnya. Format PNG banyak digunakan oleh aplikasi-aplikasi *browser*.

Seiring dengan perkembangan jaman, maka media digital menjadi semakin terkenal dan semakin baik. Namun, hal tersebut juga menyebabkan terjadinya plagiarisme dan pembajakan semakin mudah. Untuk mengatasi permasalahan tersebut maka diperlukan adanya penanda pada media digital tersebut sehingga dapat diidentifikasi apakah media tersebut asli atau tidak dan dapat diketahui siapa pemilik sesungguhnya dari media digital tersebut.

II. DASAR TEORI

1. Elliptic Curve Cryptography

   Elliptic Curve Cryptography (ECC) adalah algoritma kunci public berbasiskan struktur aljabar elliptic curve pada medan yang berhingga. Penggunaan Elliptic Curve pada kriptografi awalnya diajukan oleh Neal Koblitz dan Victor S. Miller pada tahun 1985. Elliptic Curve juga digunakan untuk algoritma faktorisasi bilangan bulat yang diaplikasikan juga untuk kriptografi.

   Elliptic Curve adalah kurva yang bentuk umumnya memenuhi persamaan sebagai berikut:

   $$y^2 = x^3 + ax + b$$

   Elliptic Curve terdefinisi untuk x dan y berelemen Real. Didefinisikan sebuah titik O(x,∞) yang merupakan titik pada *infinity*.

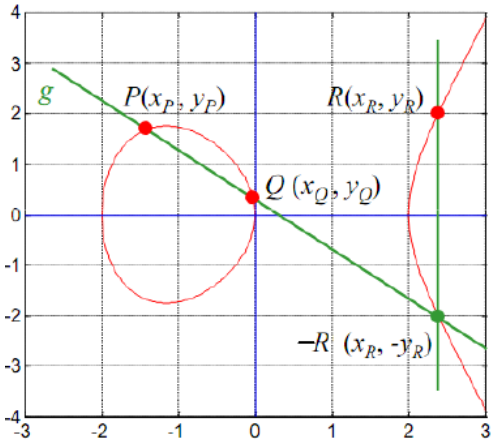
   Pada Elliptic Curve, ada beberapa operasi yang dapat dilakukan antara lain:

   - Penjumlahan Titik

     P + Q = R

     Ditarik sebuah garis g yang menghubungkan

titik P dan Q. Garis g akan memotong sebuah titik lagi pada Elliptic Curve disebut sebagai -R. Kemudian -R dicerminkan terhadap sumbu x untuk mendapatkan R.



Gambar 1. Operasi penjumlahan pada Elliptic Curve.

Persamaan garis g: $y = \lambda x + \beta$

Gradien garis g: $\lambda = \frac{y_p - y_q}{x_p - x_q}$

Perpotongan garis g dengan kurva

$$(\lambda x + \beta)^2 = x^3 + ax + b$$

Koordinat titik R:

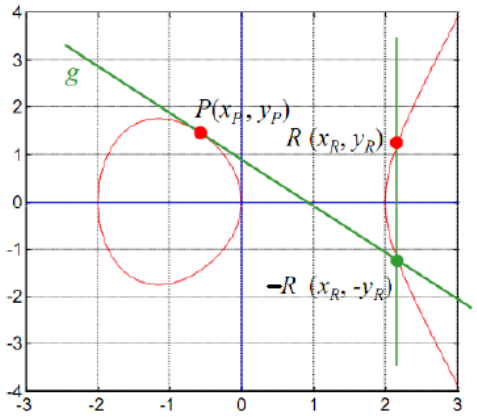
$$x_r = \lambda^2 - x_p - x_q$$

$$y_r = \lambda(x_p - x_r) - y_p$$

- Penggandaan Titik

  2P = P + P = R

  Ditarik sebuah garis g yang membentuk tangen pada titik P. Garis g akan memotong sebuah titik lagi pada Elliptic Curve disebut sebagai -R. Kemudian -R dicerminkan terhadap sumbu x untuk mendapatkan R.



Gambar 2. Operasi penggandaan pada Elliptic Curve.

Persamaan garis g: $y = \lambda x + \beta$

Gradien garis g: $\lambda = \frac{dy}{dx} = \frac{3x_p^2 + a}{2y_p}$

Perpotongan garis g dengan kurva

$$(\lambda x + \beta)^2 = x^3 + ax + b$$

Koordinat titik R:

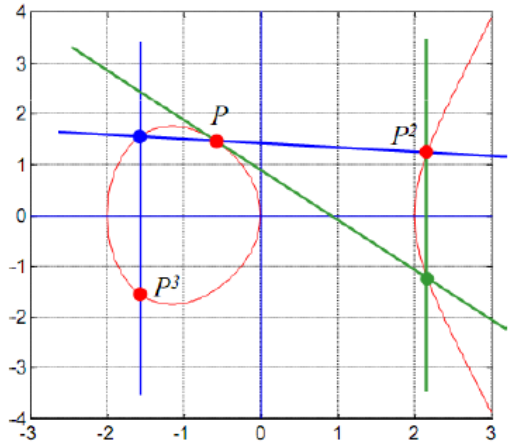
$$x_r = \lambda^2 - 2x_p$$

$$y_r = \lambda(x_p - x_r) - y_p$$

Jika yp bernilai 0 maka 2P = O.

- Pelelaran Titik

  $P^k = kP = P + P + ... + P$



Gambar 3. Operasi pelelaran pada Elliptic Curve.

Elliptic Curve memenuhi aksioma-aksioma sebagai berikut:

- Closure: semua operasi P + Q berada di dalam Elliptic Curve
- Asosiatif: (P + Q) + R = P + (Q + R)
- Elemen netral adalah O: P + O = O + P = P
- Elemen invers adalah -P: P + -P = O
- Komutatif: P + Q = Q + P

Dalam Elliptic Curve Cryptography, algoritma kunci publik lainnya, operasi dasarnya digantikan dengan operasi dasar yang berada dalam lingkup Elliptic Curve.

2. Digital Signature Algorithm

Digital Signature Algorithm (DSA) adalah standar dari *United States Federal Government standard* untuk tanda tangan digital. Hal ini diajukan oleh National Institute of Standards and Technology (NIST) pada Agustus 1991 sebagai standar dari tanda tangan digital.

Proses generasi kunci terdiri dari dua fase, yang pertama menentukan algoritma yang akan digunakan dan membentuk parameter algoritma tersebut, kemudian menghitung kunci public dan kunci privat untuk seorang pengguna.

Proses:

- Fase 1
  - Menentukan fungsi hash kriptografi yang sudah diakui, saat ini yang sering digunakan sebagai standar adalah SHA-2.
  - Menentukan panjang kunci L dan N untuk menentukan kekuatan dari kunci. Standarnya untuk nilai L merupakan kelipatan 64 berada di antara 512 sampai 1024 (inklusif).
  - Memilih bilangan prima q yang panjangnya N-bit. N harus lebih kecil dari panjang hasil hash.
  - Memilih bilangan prima yang

panjanganya L-bit dan dimodulus dengan p di mana p -1 adalah kelipatan dari q.
    - Memilih bilangan g yang jika dimodulo dengan p menghasilkan q.
    - Parameter algoritma (p, q, g) boleh dibagikan kepada pengguna di sistem.
- Fase 2
    - Memilih bilangan x dengan metode bilangan acak di mana $0 < x < q$.
    - Menghitung $y = g^x \bmod p$.
    - Kunci public adalah (p, q, g, y), kunci privat adalah x.

Proses penandatanganan dengan menggunakan DSA adalah sebagai berikut, di mana H adalah fungsi hash, dan m adalah pesan yang ingin ditandatangan:

- Memilih bilangan k dengan metode bilangan acak di mana $0 < k < q$.
- Menghitung $r = (g^k \bmod p) \bmod q$.
- Jika seandainya r = 0, ulangi dengan bilangan k acak lainnya.
- Hitung $s = k^{-1} (H(m) + xr) \bmod q$
- Jika seandainya s = 0, ulangi dengan bilangan k acak lainnya.
- Tanda tangan digitalnya adalah (r, s).

Proses verifikasi tanda tangan dengan menggunakan DSA adalah sebagai berikut:

- Tanda tangan salah jika persamaan $0 < r < q$ atau $0 < s < q$ tidak terpenuhi.
- Hitung $w = s^{-1} \bmod q$.
- Hitung $u1 = H(m) * w \bmod q$
- Hitung $u2 = r * w \bmod q$
- Hitung $v = ((g^{u1} y^{u2}) \bmod p) \bmod q$
- Tanda tangan benar jika v = r.

3. Elliptic Curve Digital Signature Algorithm

Elliptic Curve Digital Signature Algorithm (ECDSA) adalah algoritma Digital Signature Algorithm (DSA) yang memanfaatkan elliptic curve cryptography.

Proses penandatanganan dengan menggunakan ECDSA adalah sebagai berikut:

- Menentukan elliptic curve dan persamaan yang digunakan pada elliptic curve.
- Menentukan titik dasar yang berorde prima n yaitu G pada kurva.
- Membuat pasangan kunci, menentukan sebuah bilangan d_A yang secara acak dipilih di antara 1 dan n-1 inklusif. d_A merupakan kunci privat. Menentukan $Q_A = d_A * G$ dengan perkalian elliptic curve. Q_A merupakan kunci publik.
- Melakukan hash pada pesan m sehingga menghasilkan e. e = HASH(m).
- Menentukan z di mana z adalah Ln Leftmost bit dari e. Ln adalah panjang bit berorde n.
- Menentukan k dengan pembangkit bilangan acak di mana k bernilai di antara 1 dan n-1 inklusif.
- Menghitung point pada kurva (x1, y1) = k * G.
- Menghitung nilai r = x1 (mod n). Jika r bernilai 0, maka pilih ulang nilai k.
- Menghitung nilai $s = k^{-1}(\text{int}(z) + rd_A) \pmod n$. Jika s bernilai 0, pilih ulang nilai k.
- Tanda tangan digitalnya adalah (r, s).

Proses verifikasi tanda tangan dengan menggunakan ECDSA adalah sebagai berikut:

- Verifikasi apakah Q_A adalah titik pada kurva yang valid dengan mengecek Q_A tidak sama dengan O (elemen netral), mengecek Q_A berada pada kurva, dan mengecek apakah $n * Q_A = O$.
- Verifikasi apakah nilai r dan s berada di antara 1 dan n-1 inklusif. Jika tidak, maka tanda tangan digital salah.
- Menghitung e = HASH(m), di mana HASH adalah fungsi hash yang sama dengan fungsi hash yang digunakan pada saat tanda tangan.
- Menentukan z di mana z adalah L_n Leftmost bit dari e.
- Menghitung $w = s^{-1} \pmod n$.
- Menghitung $u1 = \text{int}(z)w \pmod n$.
- Menghitung $u2 = rw \pmod n$.
- Menghitung titik pada kurva (x1, y1) = $u1 * G + u2 * Q_A$.
- Tanda tangan benar jika $r = x1 \pmod n$,

Pada proses penandatanganan, k sangat disarankan dibangkitkan secara acak untuk setiap bagian pesan, karena jika tidak maka kunci privat bisa dicari dengan perhitungan. Pada bulan Desember 2010, sebuah grup yang menyebut dirinya failOverflow berhasil mendapatkan kunci privat dari perusahaan Sony untuk menandatangani console game Playstation 3.

III. IMPLEMENTASI

A. Lingkungan Implementasi

Implementasi dilakukan pada lingkungan bahasa pemrograman Java 64 bit yang dijalankan pada Sistem Operasi Windows 8 Professional 64 bit. Implementasi menggunakan *Integrated Development Environment* (IDE) berupa Eclipse Juno 4.2. Dalam pembuatannya, interface yang digunakan adalah *Graphical User Interface* (GUI) dari Java yaitu *Swing*. Dalam hal layout, menggunakan *library* yaitu *miglayout*.

Untuk membangkitkan kunci Elliptic Curve Digital Signature Algorithm menggunakan *library* dari bouncycastle. Dan juga untuk menandatangani dan memverifikasi menggunkana *library* dari bouncycastle.

B. Implementasi Penandatanganan Digital

Untuk implementasi penandatanganan digital, proses yang dilakukan adalah dengan menggenerate pasangan kunci publik dan privat kemudian dilanjutkan dengan

menandatangani citra yang sudah dibaca. Diagram alir secara umum untuk proses penandatangan digital dapat dilihat pada gambar 4.

Gambar 4. Diagram alir proses penandatanganan.

Pada saat menyimpan gambar ke dalam *file*, program menambahkan digital signature pada bagian paling bawah dari gambar dengan tag <ds></ds>. Dalam percobaan yang dilakukan penambahan tidak dapat dilakukan pada *file* gambar dengan format JPEG. Hal tersebut disebabkan karena kompresi pada format JPEG bersifat *lossy data compression* yang berarti data yang sudah dikompresi belum tentu dapat dikembalikan menjadi data semula seutuhnya sehingga hasil hash dari fungsi hash akan selalu menghasilkan hasil yang berbeda. Namun untuk format gambar lain seperti PNG tidak bermasalah karena kompresinya bersifat *lossless data compression* yang berarti data dapat dikembalikan sepenuhnya setelah dikompresi.

Berikut contoh implementasi dari penandatanganan digital yang dilakukan:

Gambar 5. Gambar Lena yang dijadikan percobaan (lena.bmp).

Kunci Privat:

S:

7ea3005bbe50ae14139f8ab8a8d9cc6f92a6e4ae1984d322

Kunci Publik:

X:

a9d7d7724be2fbb897e44ab87ec41ef4c9af06b2b4addca8

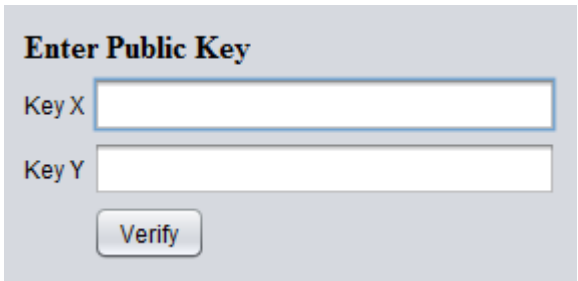
Y:

a1aed2ba44557299eb725968b5418843b6d6c1f9d6d71cf0

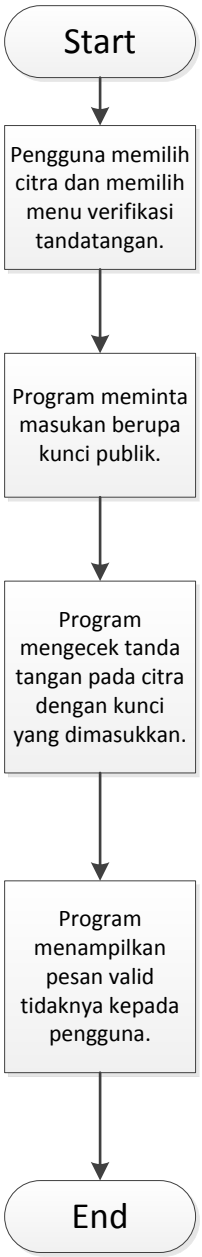
Panjang Signature 56 byte.

C. Implementasi Verifikasi Tanda Tangan Digital

Untuk implementasi verifikasi tanda tangan digital adalah dengan meminta masukkan kunci publik dari pengguna berupa koordinat titik pada sumbu X dan sumbu Y. Kemudian dilanjutkan dengan pengecekkan tanda tangan digital yang sudah ada di *file* gambar. Program akan memberitahu pengguna tentang status dari pengecekkan tanda tangan digital. Diagram alir secara umum untuk proses verifikasi tanda tangan digital dapat dilihat pada gambar 7.



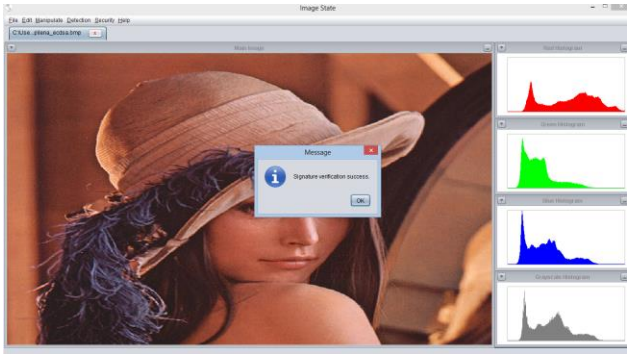
Gambar 6. GUI program meminta masukan kunci publik dari pengguna.



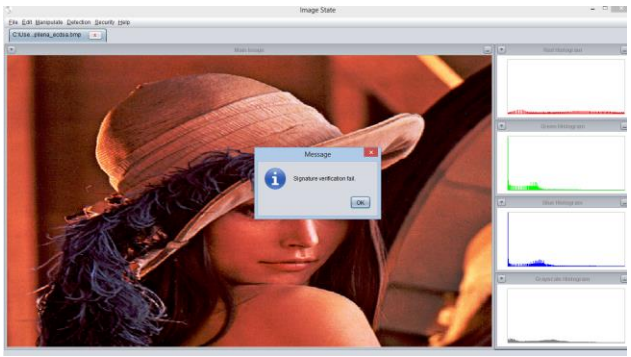
Gambar 7. Diagram alir proses verifikasi tanda tangan digital.

Pada saat proses verifikasi, program mengambil tanda tangan digital yang sudah ada di *file* di antara tag <ds></ds> dan selanjutnya menjalankan proses verifikasi.

Berikut contoh implementasi dari verifikasi tanda tangan digital yang dilakukan:



Gambar 8. Gambar Lena yang sukses diverifikasi.



Gambar 9. Gambar Lena yang gagal diverifikasi.

Pada gambar 9, tanda tangan digital gagal diverifikasi karena pada citra dilakukan equalisasi histogram sehingga keadaan citra berubah.

IV. Perbandingan Algoritma

Pada citra yang sama juga dilakukan percobaan dengan algoritma DSA yang biasa. Diperoleh hasil yang berbeda sebagai berikut:

Kunci Publik ECDSA:
X:
a9d7d7724be2fbb897e44ab87ec41ef4c9af06b2b4addca8
Y:
a1aed2ba44557299eb725968b5418843b6d6c1f9d6d71cf0
Total byte: 24 byte.

Kunci Public DSA:
y:
60cfc214db82d434b07b191ebd91e0adc3d09c01a9cbd087
5e0967acc261c9e9da512302cc00d456d33a79ffe312b473
30f4f9b78a5fb56ca351adccd632132a61c1399c86e0ed90e
237bbbb3f15351fab7beb138a6d35836f3162be88fcf058d
9fd6becf82cb4a74c965aee0398fd92040d642707917f2ee9
ac67a2aa3d88f9
Total byte: 136 byte.

Signature ECDSA:
Total byte: 56 byte.

Signature DSA:
Total byte: 46 byte.

Kunci Privat total bytenya tidak berbeda.

Jika dilakukan perbandingan maka dapat diketahui bahwa dengan menggunakan Elliptic Curve Cryptography pada DSA telah menghemat byte 102 byte dari 182 byte. Persentasenya sebesar 56%. Hal ini tentu sangat berpengaruh terutama pada implementasi pada komputer dengan memori kecil seperti smart card.

V. KESIMPULAN

Dari pembahasan di makalah ini, dapat disimpulkan bahwa Elliptical Curve Cryptography dapat menghemat jumlah memori yang diperlukan untuk melakukan enkripsi ataupun dekripsi sehingga juga menghemat komputasi dan tanda tangan digital dapat memberikan identitas pada citra ataupun media digital lainnya.

DAFTAR PUSTAKA

[1] http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/kriptografi.htm
[2] http://www.miglayout.com/
[3] http://www.bouncycastle.org/

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, 20 Mei 2013

Jordan Fernando / 13510069