Elliptic Curve dan Implementasinya pada Algoritma Tanda Tangan Digital

Vincent Kusuma dan Darmaji
Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan alam,
Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia
e-mail: darmaji@matematika.its.ac.id

***Abstrak*—Tanda tangan digital adalah skema tanda tangan untuk dokumen digital. Algoritma tanda tangan digital yang sering digunakan adalah Diffie-Hellman *Digital Signature Algorithm* dan lebih dikenal sebagai *Digital Signature Algorithm* (DSA) lalu dikembangkan menjadi *Elliptic Curve Digital SignatureA lgorithm* (ECDSA). Tugas akhir ini mengimplementasikan ECDSA untuk membuat suatu tanda tangan digital. Waktu eksekusi yang dibutuhkan oleh ECDSA dan DSA untuk membuat dan memverifikasi hampir sama, akan tetapi waktu yang dibutuhkan *brute force attack* untuk menemukan nilai dari *private key* dari ECDSA 15,5 kali lebih lama dibanding waktu yang dibutuhkan *brute force attack* untuk menemukan nilai dari *private key* dari DSA. Sehingga ECDSA memiliki keamanan yang lebih baik daripada DSA tanpa mengorbankan lama waktu eksekusi.**

***Kata Kunci*— Tanda tangan, Tanda tangan Digital, Algoritma Tanda Tangan Digital, DSA, *Elliptic Curve*, ECDSA, Keamanan**

I. PENDAHULUAN

SEBELUM era digital, sebuah pesan dikirim dalam bentuk kertas dan tanda tangan dibubuhkan oleh pengirim pesan untuk memberi kepastian atas keaslian pesan tersebut. Tanda tangan konvensional yang berupa *coretan tangan* tidak dapat digunakan untuk memastikan keaslian pesan digital meski diproses menjadi bentuk citra digital karena kita dapat dengan mudah menduplikasi suatu citra dengan menggunakan aplikasi komputer. Oleh karena itu dibuatlah suatu skema tanda tangan untuk dokumen digital yaitu tanda tangan digital.

Salah satu algoritma tanda tangan digital yang sering digunakan adalah *Diffie-Hellman Digital Signature Algorithm* dan lebih dikenal sebagai *Digital Signature Algorithm* (DSA). Algoritma ini menggunakan *prime finite field* dengan bilangan prima yang cukup besar sehingga menyulitkan pihak yang ingin menduplikasi tanda tangan yang dihasilkan. Sayangnya *prime finite field* kurang fleksibel sehingga pemilihan *key* kurang beragam. *Elliptic Curve Digital Signature Algorithm* (ECDSA) merupakan pengembangan dari DSA. ECDSA menggunakan *elliptic curve* atas suatu *prime finite field* sehingga lebih fleksibel dalam proses menentukan *key*.

Makalah ini bertujuan untuk melakukan pengujian terhadap proses tanda tangan digital menggunakan DSA dan ECDSA. Proses pada tanda tangan digital tersebut akan diuji menggunakan 10 bilangan prima acak dan sebuah dokumen dengan tipe *text* berekstensi txt. Diharapkan penggunaan *ECDSA* dapat meningkatkan keamanan tanda tangan dibandingkan dengan DSA tanpa mengorbankan waktu proses penanda-tanganan dan verifikasi.

II. DASAR TEORI

A. *Digital Signature Algorithm (DSA)*

DSA merupakan salah satu algoritma untuk tanda tangan digital yang menggunakan *prime finite field* untuk menentukan *key*. Ada tiga tahapan pada DSA yaitu tahap menentukan *key*, tahap penanda-tanganan, dan tahap verifikasi [1]. Tahapan-tahapan dari DSA dapat dilihat pada Gambar 1 - 3.

Tahap menentukan *key*:

1. Pilih suatu bilangan prima q.
2. Cari bilang bilangan prima p dengan $p \equiv 1 \bmod q$.
3. Cari suatu bilangan g yang merupakan generator dari subgrup F_p^* dengan order q dengan memilih secara acak $h \in \{1,2,\ldots,p-1\}$ lalu hitung $g = h^{\frac{p-1}{q}} \bmod p$. Jika $g = 1$, pilih h yang lain lalu ulangi.
4. Pilih $x \in \{1,2,\ldots,q-1\}$ sebagai *private key*.
5. Hitung $y = g^x \bmod p$

Gambar 1. Tahap menentukan *key* pada DSA

Pada tahap ini diperoleh y sebagai *public key* dan x sebagai *private key*.

Tahap penanda-tanganan.
Masukan: pesan m dan *private key* x.
Luaran: tanda tangan (r,s)

1. Pilih secara acak $k \in \{1,2,\ldots,q-1\}$.
2. Hitung $t = g^k \bmod p$.
3. Definisikan fungsi $f: F_p^* \to F_q$ dengan $f(x) = x \bmod q$.
4. Hitung $r = f(t)$, jika $r = 0$ ulangi langkah 1.
5. Hitung $s = H(m)$, dimana H merupakan *hash function*.
6. Hitung $s = (s + xr)k^{-1} \bmod q$, jika $s = 0$ ulangi langkah 1.

Gambar 2. Tahap penanda-tanganan pada DSA

Tahap ini menghasilkan tanda tangan yang berupa pasangan bilangan bulat (r,s) atas suatu *private key* x dan pesan m.

Tahap verifikasi.
Masukan: pesan m, *public key* y dan tanda tangan (r,s)

1. Tolak jika $r,s \notin \{1,2,\ldots,q-1\}$.

2. Hitung $s = H(m)$.
3. Hitung $u_1 = es^{-1} \bmod q$.
4. Hitung $u_2 = rs^{-1} \bmod q$.
5. Hitung $t = g^{u_1} y^{u_2} \bmod p$.
6. Terima jika dan hanya jika $f(t) = r$.

Gambar 3. Tahap verifIkasi pada DSA

Tahap ini menentukan keaslian tanda tangan digital yang dibubuhkan pada pesan m dengan cara melakukan cek pada nilai $f(t)$ yang seharusnya bernilai sama dengan r.

B. Elliptic Curve

Sebelum membahas ECDSA, terlebih dahulu dibahas mengenai struktur dari *elliptic curve*.

Definisi 2.1. Definisi persamaan *elliptic curve* [2].

Sebuah *elliptic curve* E atas *field* F didefinisikan oleh persamaan

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,\ a_i \in F \quad (1)$$

Dengan $\Delta \neq 0$ dimana Δ adalah diskriminan dari E yang didefinisikan oleh persamaan

$$\Delta = -d_2^2 d_8 - 8d_4^3 - 27d_6^2 + 9d_2 d_4 d_6 \quad (2)$$

Dengan

$$\left.\begin{aligned} d_2 &= a_1^2 + 4a_2 \\ d_4 &= 2a_4 + a_1a_3 \\ d_6 &= a_3^2 + 4a_6 \\ d_8 &= a_1^2 a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \end{aligned}\right\} \quad (3)$$

Notasi $E(F)$ didefinisikan sebagai himpunan titik-titik $(x, y) \in F^2$ yang memenuhi persamaan (1), bersama dengan "*point at infinity*" yang dinotasikan dengan *O*. Sedangkan notasi $\#E(F)$ didefinisikan sebagai banyaknya elemen pada $E(F)$.

Definisi 2.2. Penjumlahan titik pada *elliptic curve* untuk $E: y^2 = x^3 + ax + b,\ a, b \in F_p$ dan $p > 3$ didefinisikan sebagai berikut.

1. $P + O = O + P = P$, untuk setiap $P \in E(F_p)$, O berlaku sebagai elemen netral pada $E(F_p)$
2. diberikan $P \in E(F_p)$ dengan $P = (x, y)$, maka $-P$ adalah invers dari P yang didefinisikan dengan $-P = (x, -y)$ sehingga $P + (-P) = (-P) + P = O$.
3. Diberikan $P, Q \in E(F_p)$ dengan $P = (x_1, y_1)$ dan $Q = (x_2, y_2)$ dengan $P \neq \pm Q$. Maka $P + Q = (x_3, y_3)$ dimana $x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2$ dan $y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1$
4. Diberikan $P \in E(F_p)$ dengan $P = (x_1, y_1)$ maka $2P = (x_3, y_3)$, dimana $x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1$ dan $y_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1$

$E(F_p)$ bersama dengan operasi $+$ membentuk suatu grup.

C. Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA merupakan algoritma hasil dari pengembangan DSA. Algoritma ini menggunakan *elliptic curve* atas *prime finite field* untuk menentukan *key*. Layaknya DSA, ada tiga tahapan pada ECDSA yaitu tahap menentukan *key*, tahap penanda-tanganan, dan tahap verifikasi [1]. Tahapan-tahapan dari ECDSA dapat dilihat pada Gambar 4 - 6.

Tahap menentukan key.
1. Pilih suatu bilangan prima p.
2. Cari bilang bilangan prima q dengan $\#E(F_p) \equiv 0 \bmod q$.
3. Cari suatu titik G yang merupakan generator dari subgrup $E(F_p)$ dengan order q dengan memilih secara acak $H \in E(F_p)$ lalu hitung $G = \frac{\#E(F_p)}{q} H$. Jika $G = O$, pilih H yang lain lalu ulangi.
4. Pilih $x \in \{1, 2, \ldots, q-1\}$ sebagai *private key*.
5. Hitung $Y = xG$

Gambar 4. Tahap menentukan *key* pada ECDSA

Tahap menentukan key ini diperoleh Y sebagai *public key* dan x sebagai *private key*.

Tahap penanda-tanganan.
Masukan: pesan m dan *private key* x.
Keluaran: tanda tangan (r, s)
1. Pilih secara acak $k \in \{1, 2, \ldots, q-1\}$.
2. Hitung $T = kG$.
3. Definisikan fungsi $f: E(F_p) \to F_q$ dengan $f(P) = x(P) \bmod q$, dimana $x(P)$ adakah koordinat-x dari titik P.
4. Hitung $R = f(T)$, jika $r = 0$ ulangi langkah 1.
5. Hitung $e = H(m)$, dimana H merupakan *hash function.*
6. Hitung $s = (e + xr)k^{-1} \bmod q$, jika $s = 0$ ulangi langkah 1.

Gambar 5. Tahap penanda-tanganan pada ECDSA

Tahap penanda-tanganan ini menghasilkan tanda tangan yang berupa pasangan bilangan bulat (r, s) atas *private key* x dan pesan m.

Tahap verifikasi.
Masukan: pesan m, *public key* Y serta tanda tangan (r, s)
1. Tolak jika $r, s \notin \{1, 2, \ldots, q-1\}$.
2. Hitung $e = H(m)$.
3. Hitung $u_1 = es^{-1} \bmod q$.
4. Hitung $u_2 = rs^{-1} \bmod q$.
5. Hitung $T = u_1G + u_2Y$.
6. Terima jika dan hanya jika $f(T) = r$.

Gambar 6 Tahap verifikasi pada ECDSA

Tahap verifikasi ini menentukan keaslian tanda tangan digital yang dibubuhkan pada pesan m dengan cara melakukan cek pada nilai $f(T)$ yang seharusnya bernilai sama dengan r.

III. UJI COBA

Pada tahap ini akan dilakukan uji coba untuk menguji performa proses tanda tangan digital yang dihasilkan dan keamanan pada DSA dan ECDSA dengan 10 kali proses tanda tangan menggunakan 10 bilangan prima acak dan sebuah

dokumen dengan tipe *text* dan berekstensi txt. Evaluasi dilakukan dengan membandingkan waktu eksekusi dari proses penanda-tanganan, verifikasi dan *brute force attack* pada DSA dan ECDSA.

A. Brute Force Attack Pada DSA dan ECDSA

Pada DSA, g dan y berlaku sebagai *public key*, padahal $y = g^x \bmod p$. *Brute force attack* pada *discrete log problem* dari DSA dapat dilakukan dengan cara melakukan cek pada nilai g^n, untuk $n = 1,2,\dots,q$ hingga ditemukan nilai n yang memenuhi $g^n = y$.

Serupa dengan kasus pada DSA, *brute force attack* pada *discrete log problem* dari ECDSA dapat dilakukan dengan cara melakukan cek pada nilai nG, untuk $n = 1,2,\dots,q$ hingga ditemukan nilai n yang memenuhi $nG = Y$. *Brute force attack* pada DSA memiliki kompleksitas $O(q)$ karena untuk melakukan cek pada $g^{n+1}, g^{n+1} = g^n . g$ dimana g^n merupakan hasil dari pengecekan sebelumnya. Sedangkan *brute force attack* pada ECDSA memiliki kompleksitas $O(q \log q)$ karena penjumlahan titik pada *elliptic curve* membutuhkan perhitungan invers modulo. Untuk memaksimalkan waktu eksekusi *brute force attack* maka dipilih nilai *private key* yang maksimal yaitu $q - 1$

B. Lingkungan Uji Coba

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam uji coba perangkat lunak ini dapat dilihat pada Tabel 1.

Tabel 1.
Lingkungan Pelaksanaan Uji Coba

Perangkat	Spesifikasi
Perangkat keras	Prosesor: Intel® Pentium® CPU B950 @ 2.10GHz 2.10GHz Memori : 4.00 GB
Perangkat lunak	Sistem Operasi: Microsoft Windows 7 Professional 64-bit Perangkat Pengembang: NetBeans IDE 7.3.1 Perangkat Pembantu: Notepad

C. Parameter Uji Coba

Pada bagian ini akan dijelaskan nilai-nilai yang menjadi parameter pada hasil uji coba. Semua parameter pada hasil uji coba ini diturunkan dari 3 parameter utama perhitungan performa pada tugas akhir ini yaitu waktu eksekusi proses penanda-tanganan, waktu eksekusi proses verifikasi , dan waktu eksekusi proses *brute force attack* dalam satuan nanodetik. Daftar lengkap seluruh nilai parameter ini dapat dilihat pada Tabel 2.

D. Hasil Uji Coba

Pada skenario ini dilakukan uji coba untuk menghitung performa dari proses tanda tangan digital menggunakan DSA dan ECDSA. Hasil performa proses tanda tangan digital DSA akan dibandingkan dengam hasil performa proses tanda tangan digital ECDSA dengan masukan yang sama. Masukan yang digunakan pada skenario uji coba ini adalah dokumen dengan tipe *text* yang berekstensi txt dan sebuah bilangan prima. Hasil percobaan dari skenario ini dapat dilihat pada Tabel 3.

Tabel 2.
Parameter Uji Coba

Nama Parameter	Deksripsi
TS_{DSA}	Rata - rata waktu eksekusi dari 10 kali proses penanda-tanganan dengan DSA
TV_{DSA}	Rata - rata waktu eksekusi dari 10 kali proses verifikasi dengan DSA
TB_{DSA}	Rata - rata waktu eksekusi dari 10 kali proses brute force attack pada DSA
TS_{ECDSA}	Rata - rata waktu eksekusi dari 10 kali proses penanda-tanganan dengan ECDSA
TV_{ECDSA}	Rata - rata waktu eksekusi dari 10 kali proses verifikasi dengan ECDSA
TB_{ECDSA}	Rata - rata waktu eksekusi dari 10 kali proses brute force attack pada ECDSA

Tabel 3.
Hasil Uji Coba

Parameter	Waktu eksekusi yang dibutuhkan(nanodetik)
TSDSA	2760531,5
TVDSA	534844,5
TBDSA	126327693711,4
TSEC	2691035,7
TVEC	532890,1
TBEC	1957412198801,4

Dari hasil uji coba ini dapat dilihat bahwa rata - rata waktu eksekusi proses penanda-tanganan dengan ECDSA adalah 2691035,7 nanodetik (0,003 detik), hasil ini hampir sama dengan rata - rata waktu eksekusi proses penanda-tanganan dengan DSA yaitu 2760531,5 nanodetik (0,003 detik). Juga dapat dilihat bahwa rata - rata waktu eksekusi proses verifikasi dengan ECDSA adalah 532890,1 nanodetik (0,0005 detik), hasil ini hampir sama dengan rata - rata waktu eksekusi proses verifikasi dengan DSA yaitu 534844,5 nanodetik (0,0005 detik). Sedangkan rata - rata waktu eksekusi proses *brute force attack* pada ECDSA adalah 1957412198801,4 nanodetik (1957,412 detik), hasil ini 15,5 kali lebih baik dari rata - rata waktu eksekusi proses *brute force attack* pada DSA yaitu 126327693711,4 nanodetik (126,328 detik).

IV. KESIMPULAN

Dari hasil uji coba dapat diambil kesimpulan bahwa performa tanda tangan digital yang dihasilkan oleh ECDSA hampir sama dengan performa tanda tangan digital yang dihasilkan oleh DSA. Untuk proses penanda-tanganan rata-rata membutuhkan 0,003 detik, dan untuk proses verifikasi membutuhkan waktu 0,0005 detik

Dari segi keamanan, *brute force attack* pada ECDSA rata - rata membutuhkan waktu 1957412198801,4 nanodetik (1957,412 detik), 15,5 kali lebih baik daripada *brute force attack* pada DSA yang rata – rata membutuhkan waktu 126327693711,4 nanodetik (126,328 detik). Keamanan dapat ditingkatkan dengan memperbesar bilangan prima, Oleh

karena itu, perbaikan yang bisa dilakukan adalah dengan memperbaiki metode pencarian bilangan prima sehingga keamanan pada DSA maupun ECDSA dapat ditingkatkan.

DAFTAR PUSTAKA

[1] Blake, I.F., Seroussi, G., Smart, N.P., 2005, *Advances in Elliptic Curve Cryptography*, Cambridge University Press, New York

[2] Hankerson, D., Vanstone, S., Menezes, A.J., 2004, *Guide to Elliptic Curve Cryptography*, Springer-Verlag, New York.