BAB V
PENUTUP

5.1. Kesimpulan

Berdasarkan penjelasan dan hasil pengujian, dapat disimpulkan bahwa *plaintext* yang akan dienkripsi dipotong menjadi beberapa bagian. Pemotongan *plaintext* tergantung pada panjang kunci yang digunakan dan banyaknya karakter dalam *plaintext*. Jika panjang kunci '*nkunci*' bit maka *plaintext* akan dipotong untuk setiap $\left\lfloor \frac{nkunci}{8} - 1 \right\rfloor$ karakter. Setiap potongan *plaintext* direpresentasikan menjadi nilai numerik dan selanjutnya direpresentasikan menjadi titik kurva *elliptic* (P_M). Kemudian dienkripsi menjadi *chipertext pair of points* (P_C) menggunakan *public key* (β) penerima.

$$P_1 = k.G_E$$

$$P_2 = P_M + k.\ \beta$$

$$P_C = (P_1\ , P_2) = (k.G_E\ , P_M + k.\ \beta\ ).$$

k adalah bilangan bulat yang dipilih secara random oleh pengirim. Untuk mendapatkan *plaintext,* penerima perlu mendekripsi *chipertext pair of points* menggunakan *private key* (V) miliknya dan dihasilkan titik kurva *elliptic* (P_M).

$$M_1 = V.P_1$$

$$P_M = P_2\ - M_1$$

Kemudian mengembalikan representasi titik menjadi *plaintext.*

Kekuatan ElGamal ECC tergantung pada panjang kunci yang digunakan. Semakin panjang kuncinya, maka akan semakin aman. Selain itu, pemilihan parameter-parameter domain ElGamal ECC juga berpengaruh terhadap tingkat keamanan. Parameter-parameter tersebut dipilih sedemikian sehingga dihasilkan order *basic point* yang terbesar. Semakin besar order *basic point*, maka pilihan dalam menentukan *private key* semakin banyak. Akibatnya, kemungkinan *cryptanalist* untuk mendapatkan *private key* semakin kecil.

Proses perhitungan dalam algoritma ElGamal ECC membutuhkan waktu yang lebih lama dibandingkan dengan *public key algorithm* yang lain. Tetapi

tingkat keamanan yang dihasilkan algoritma ElGamal ECC lebih aman dibandingkan *public key algorithm* yang lain.

Berdasarkan perkembangan dan kemampuan teknologi komputasi saat ini, rekomendasi panjang kunci yang digunakan minimal 160 bit. Implementasi 160 bit ElGamal ECC memiliki tingkat keamanan yang setara dengan 1024 RSA atau DSA.

5.2. Saran

Setelah mempelajari dan mengimplementasikan algoritma ElGamal ECC, penulis memiliki beberapa saran yang dapat dijadikan sebagai bahan pertimbangan bagi pembaca dalam mengembangkan atau melakukan penelitian tentang kriptografi kurva *elliptic*.

1. Implementasi ElGamal ECC ini menggunakan persamaan kurva *elliptic* $y^2 = x^3 + Ax + B \pmod{p}$ atas F_p. Pembaca dapat mengembangkan dan melakukan penelitian pada persamaan kurva *elliptic* yang lebih kompleks dan lapangan yang berbeda, seperti lapangan karakteristik 2 (F_2^m).
2. Pembaca dapat melakukan penelitian tentang implementasi kriptografi kurva *elliptic* untuk tanda tangan digital dan protokol pertukaran kunci. Seperti ECDSA dan Diffie-Hellman ECC.
3. Melakukan penelitian tentang algoritma yang efisien untuk memecahkan ECDLP atau algoritma yang dapat membuka *chipertext pair of point* tanpa mengetahui *private key*.
4. Mengimplementasikan algoritma ElGamal ECC pada *software* lain, seperti *Visual Basic, Delphi, C++, Pascal* dan lain sebagainya.
5. Implementasi gabungan dari skema enkripsi, tanda tangan digital dan protokol pertukaran kunci dalam skala sistem yang lebih luas. Seperti transaksi *online*, *internet banking*, lembaga intelijen atau militer dan lain sebagainya.