Membangkitkan *Signature* dengan Algoritma ElGamal Tanpa Mengetahui Kunci Rahasia

Satrio Adi Rukmono – NIM : 13506070
Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail: r.satrioadi@gmail.com

Abstrak

Algoritma ElGamal adalah algoritma enkripsi kunci asimetris untuk kriptografi kunci publik yang berlandaskan pada algortima pertukaran kunci Diffie-Hellman. Keamanan algoritma ini terletak pada sulitnya menghitung logaritma diskrit. Pada makalah ini akan ditunjukkan bahwa pada kasus-kasus tertentu *signature* dapat dibangkitkan tanpa perlu memecahkan persoalan logaritma diskrit. Hal ini dapat dilakukan jika pemilihan parameter publik tidak dilakukan dengan berhati-hati.

Keamanan algoritma ElGamal bergantung pada parameter-parameter *p* dan *g*. Dengan beberapa informasi tambahan mengenai parameter *g*, cipherteks dapat dibangkitkan tanpa perlu membangkitkan kunci privat terlebih dahulu. Hal ini memungkinkan dibuatnya *trapdoor* saat menentukan parameter *p* dan *g* dalam sebuah kelompok.

Teknik ini tidak berlaku untuk semua kasus dalam algoritma ElGamal, tetapi menunjukkan bahwa penggunaan algoritma ElGamal harus dilakukan dengan seksama.

***Kata kunci**:* kriptografi, ElGamal, Diffie-Hellman.

1. Pendahuluan

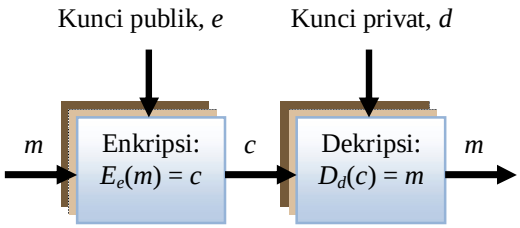
Kriptografi simetri menggunakan kunci yang sama untuk enkripsi dan dekripsi. Hal ini mengimplikasikan kedua pihak yang berkomunikasi saling mempercayai, karena kedua pihak harus menjaga kerahasiaan kunci. Masalah utama dalam kriptografi simetri adalah cara mendistribusikan kunci. Distribusi kunci dari pengirim pesan kepada penerima pesan melalui saluran publik (pos, telepon, dan lain-lain) tentu tidak aman, karena memungkinkan terjadinya penyadapan saat pengiriman kunci. Oleh karena itu dibutuhkan saluran komunikasi yang lebih terpercaya, yang umumnya lebih lambat dan lebih mahal.

Hal tersebut mendorong timbulnya kriptografi nirsimetri (atau kriptografi kunci-publik) pada akhir tahun 1975 oleh Diffie dan Hellman, yang memungkinkan pengguna berkomunikasi dengan aman tanpa perlu berbagi kunci rahasia. Kunci untuk enkripsi berbeda dengan kunci untuk dekripsi, maka kunci untuk enkripsi dapat disebarkan dengan bebas sedangkan kunci untuk dekripsi hanya diketahui oleh penerima pesan. Oleh karena itu kunci untuk enkripsi disebut juga kunci publik, sedangkan kunci untuk dekripsi disebut dengan kunci privat.

Kriptografi kunci-publik ini memberi dua keuntungan: yang pertama, tidak dibutuhkan saluran komunikasi kedua untuk distribusi kunci. Kunci publik dapat dikirimkan melalui saluran yang sama dengan saluran pengiriman pesan yang umumnya tidak aman, sebab dengan mengetahui kunci publik tidak berarti dapat melakukan dekripsi. Kedua, jumlah kunci dapat ditekan – komunikasi dengan banyak orang hanya membutuhkan dua buah kunci, yaitu kunci publik dan kunci privat. Dalam kriptografi kunci simetri, untuk berkomunikasi dengan banyak orang dibutuhkan kunci sebanyak jumlah orang yang terlibat dalam komunikasi.

Berbeda dengan kriptografi simetri yang berdasar pada substitusi dan permutasi, kriptografi kunci-publik berdasar pada fungsi

matematika. Kriptografi simetri mengandalkan panjang kunci, sedangkan kriptografi kunci-publik mengandalkan sulitnya memecahkan masalah matematis.



Gambar 1. Skema kriptografi kunci-publik. Plainteks *m* dienkripsikan dengan kunci *e*, namun cipherteks *c* didekripsikan dengan kunci yang berbeda, yaitu *d*.

Salah satu contoh algoritma kriptografi kunci-publik adalah algortima ElGamal, yang memanfaatkan sulitnya menghitung logaritma diskrit.

2. Algortima ElGamal

Algoritma ElGamal dibuat oleh Taher ElGamal pada tahun 1984. Algoritma ini awalnya digunakan untuk *digital signature*, namun kemudian dimodifikasi untuk keperluan enkripsi-dekripsi. Keamanan algoritma ini terletak pada sulitnya menghitung logaritma diskrit. Algoritma ElGamal untuk *digital signature* adalah sebagai berikut:

Tentukan parameter publik: bilangan prima p, dan g di mana $g < p$.

Tentukan kunci privat, bilangan acak x di mana $x < p$ dan kunci publik, yaitu $y = g^x \pmod p$.

Signature (a, b) untuk m dapat ditentukan dengan rumus

$$a \equiv g^k \pmod p$$
$$b \equiv (m - xa)k^{-1} \pmod{p - 1}$$

dengan k adalah bilangan acak dengan $k \in \{0, \ldots, p\text{-}2\}$.

Selanjutnya untuk verifikasi bahwa pasangan *signature* (a, b) valid untuk m digunakan persamaan

$$g^m \equiv a^b y^a \pmod p$$

Bilangan prima p harus cukup besar agar penghitungan logaritma diskrit menjadi sulit dilakukan dan p-1 harus memiliki setidaknya satu faktor prima untuk membuat algoritma Pohlig-Hellman tidak berlaku.

3. Serangan-serangan pada Algoritma ElGamal

3.1. Pemilihan parameter yang lemah

Keamanan algoritma ElGamal sangat bergantung pada parameter p dan g. Jika ada informasi tambahan mengenai g, maka cipherteks untuk m dapat ditentukan tanpa mengetahui kunci privat x.

Misalkan $p - 1 = rw$ dengan y sebagai kunci publik. Jika terdapat $\beta = sw$ dengan $0 < s < r$ dan bilangan bulat t di mana $\beta^t \equiv g \pmod p$ maka *signature* (a, b) untuk m dapat ditemukan.

Hal ini dapat dibuktikan sebagai berikut: persamaan

$$g^{wz} \equiv y^w \pmod p$$

dapat diselesaikan untuk z. Hal ini didapat dari $p - 1 = rw$ sehingga g^w memiliki orde r. Maka logaritma diskrit dapat dihitung dengan algoritma Pohlig-Hellman, sehingga z dapat ditemukan.

Kini *signature* (a, b) dapat dihitung dengan

$$a = \beta$$
$$b \equiv t(m - swz) \pmod{p - 1}$$

sebab

$$a^b y^a \equiv (\beta^t)^{(m-swz)} y^{sw}$$
$$\equiv g^{m-swz}\, g^{swz} \equiv g^m \pmod p.$$

Selanjutnya, jika g habis membagi $p - 1$ maka *signature* untuk m dapat dibangkitkan. Misalkan

$$\beta = \frac{(p-1)}{g} \text{ dan } t = \frac{(p-3)}{2},$$

maka

$$\beta^t \equiv (-1)\beta^{-1} \equiv g \pmod{p}.$$

Dengan demikian, sesuai uraian sebelumnya, *signature* untuk m dapat dibangkitkan tanpa mengetahui kunci privat x.

Hal ini menunjukkan bahwa jika parameter g tidak dipilih secara hati-hati, *signature* ElGamal untuk setiap m dapat ditentukan tanpa mengetahui kunci privat x. Memilih $g = 2$ bukanlah hal yang baik sebab dengan demikian pemilihan parameter p tidak berpengaruh terhadap pembangkitan *signature*.

Metode ini dapat berhasil dengan memilih a yang tepat dari himpunan $\beta^t \equiv g \pmod{p}$, yang memudahkan penghitungan logaritma diskrit.

3.2. *Trapdoor* untuk algoritma ElGamal

Seringkali dalam sebuah kelompok digunakan parameter p dan g yang sama agar kunci publik menjadi lebih pendek. Selain itu, jika eksponen sudah dihitung sebelumnya, penghitungan dan verifikasi *signature* dapat dilakukan jauh lebih cepat. Namun hal ini membuka peluang bagi pihak yang berwenang dalam menentukan kedua parameter tersebut untuk membuat *trapdoor* agar ia dapat membangkitkan *signature* seluruh pesan yang beredar dalam kelompok tersebut.

Pihak yang berwenang tersebut dapat bermain 'nakal' dengan menentukan parameter p dan g sedemikian rupa sehingga ia mengetahui nilai β dan t yang dapat digunakan untuk membangkitkan *signature*. Hal ini dapat dilakukan dengan metode sebagai berikut:

Nilai-nilai g, β, dan t ditentukan dari nilai parameter prima p. Pertama, tentukan r dan w sehingga $p - 1 = rw$. Jika r cukup kecil, maka dengan mudah dapat ditemukan $\beta = cw$ sehingga β memenuhi persyaratan yang sama dengan parameter g. Selanjutnya t dipilih sedemikian rupa sehingga FPB(t, $p - 1$) = 1 dan g dihitung dengan $g = \beta^t$. Namun, jika r terlalu kecil, maka tidak ada nilai cw yang memenuhi.

Selain metode di atas, masih ada metode lain yang merupakan kebalikannya, yaitu nilai-nilai p, β, dan t ditentukan dari nilai parameter g.

4. Pencegahan Terhadap Serangan pada Algoritma ElGamal

Serangan-serangan yang telah dijelaskan pada bab 3 dapat dihindari hanya jika pada *signature* (a, b), nilai a tidak habis dibagi oleh q, yaitu faktor prima dari $p - 1$. Kondisi ini harus selalu diperiksa saat verifikasi. Selain itu, sangat kecil peluang bahwa a habis dibagi q jika dibangkitkan secara acak, maka jika p dan g memenuhi persyaratan standar untuk algoritma ElGamal, hampir pasti bahwa a tidak habis dibagi q. Kondisi ini sudah tercakup dalam *digital signature standard* (DSS) sehingga serangan di atas tidak berlaku untuk *signature* yang memenuhi DSS.

Selain itu, untuk mencegah penyalahgunaan kekuasaan oleh pihak yang menentukan parameter-parameter dalam algoritma ElGamal, pihak tersebut dapat diharuskan untuk menggunakan algoritma khusus dalam membangkitkan parameter p dan q, misalnya algoritma oleh NIST yang juga sudah tercakup dalam DSS. Dengan algoritma ini, publik dapat melakukan verifikasi bahwa parameter-parameter p dan g benar-benar dibangkitkan dengan algoritma terserbut atau tidak.

Metode lain untuk menghindari serangan pada algoritma ElGamal tentu saja adalah modifikasi pada persamaan yang digunakan untuk membangkitkan *signature* dan verifikasi *signature*. Terdapat banyak variasi untuk memodifikasi persamaan yang digunakan pada algoritma ElGamal, namun variasi persamaan tersebut harus dilakukan dengan sangat berhati-hati, sebab pemilihan variasi yang tidak tepat justru akan menimbulkan masalah baru.

5. Kesimpulan

Makalah ini menunjukkan bahwa pembangkitan *signature* menggunakan algoritma ElGamal terkadang dapat dilakukan tanpa mengetahui kunci privat atau kunci rahasia. Serangan-serangan yang ada dapat dicegah dengan membatasi nilai *signature* terhadap parameter-parameter yang digunakan. Dengan demikian makalah ini tidak menunjukkan bahwa algoritma ElGamal sudah obsolet, melainkan justru menunjukkan bahwa pemilihan parameter harus dilakukan dengan seksama untuk mencegah kemungkinan-kemungkinan serangan yang ada.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] Kuhn, Markus. (2009). Introduction to Security. Cambridge University Press.

[3] Miyaji, Atsuko. (1997). A Trapdoor Generating Algorithm Over Elliptic Curve ElGamal Signature.

[4] Nguyen, Phong Q. (2008). Public-Key Cryptanalysis.

[5] Pohlig, S. C. dan Hellman, M. E. (1978). An Improved Algorithm for Computing Logarithms over GF(p) and Its Cryptographic Significance. IEEE Trans. Inform. Theory.