PENYANDIAN PESAN MENGGUNAKAN KOMBINASI ALGORITMA RSA YANG DITINGKATKAN DAN ALGORITMA ELGAMAL

Jafarudin Firdaus[1], Rini Marwati[2], Sumanang Muhtar Gozali[3]

[1, 2, 3] Departemen Pendidikan Matematika FPMIPA UPI

*Surel: jafarudinfirdaus@gmail.com

ABSTRAK. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Algoritma kriptografi yang populer saat ini adalah algoritma RSA yang didasarkan pada masalah pemfaktoran bilangan bulat dan algoritma ElGamal yang didasarkan pada masalah logaritma diskrit. Pada makalah ini, diusulkan sebuah algoritma baru kriptografi yang menggabungkan algoritma ElGamal dengan algoritma RSA yang ditingkatkan menggunakan tiga bilangan prima.

Kata Kunci: Kriptografi, RSA, RSA yang ditingkatkan, ElGamal

ABSTRACT. Cryptography is the art and science of keeping the security of messages. Nowadays, the most popular cryptography algorithms are the RSA algorithm which is based on integer factorization problem and the ElGamal algorithm which is based on discrete logarithm problem. In this paper, a new cryptography algorithm is proposed by combining the ElGamal algorithm and the enhanced RSA algorithm that use three prime numbers.

Keywords: Cryptography, RSA, Enhanced RSA, ElGamal

1. PENDAHULUAN

Salah satu media untuk berkomunikasi adalah melalui tulisan. Tulisan berfungsi untuk menyampaikan pesan. Pesan yang terkandung dalam tulisan bisa berisi informasi rahasia maupun tidak. Informasi rahasia dapat terjaga keamanannya jika disampaikan oleh pemberi pesan secara langsung kepada penerima pesan. Namun jika pesan disampaikan secara tidak langsung melalui pihak ketiga maka informasi rahasia pada pesan tidak terjamin keamanannya.

Salah satu cara yang dapat ditempuh untuk mengamankan pesan adalah dengan mengubahnya menjadi sandi-sandi yang sulit dibaca dan hanya bisa dibaca oleh pihak tertentu, metode ini disebut kriptografi.

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. Crypto berarti *secret* (rahasia) dan graphia berarti *writing* (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain [1].

Algoritma kriptografi terbagi menjadi dua jenis yaitu algoritma simetris dan algoritma asimetris. Algoritma simetris adalah suatu algoritma yang menggunakan kunci enkripsi sama dengan kunci dekripsi. Algoritma asimetris adalah suatu algoritma yang menggunakan kunci berbeda pada proses enkripsi dan dekripsi. Algoritma asimetris yang populer saat ini adalah algoritma RSA dan algoritma ElGamal. Algoritma RSA terkenal keamanannya karena masalah pemfaktoran bilangan bulat, sedangkan algoritma ElGamal terkenal keamanannya karena masalah logaritma diskrit.

Berdasarkan uraian di atas, diketahui bahwa kedua algoritma tersebut aman. Dalam penelitan ini, dilakukan penggabungan dua algoritma asimetris tersebut agar diperoleh suatu algoritma baru yang bisa digunakan untuk menyandikan pesan yaitu dengan meningkatkan kecepatan algoritma RSA lalu digabungkan dengan algoritma ElGamal.

2. KRIPTOGRAFI RSA

Algoritma kriptografi RSA dibuat oleh tiga orang peneliti MIT (*Massachussets Institute of Technology*) pada tahun 1976, yaitu Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman. Algoritma ini terdiri atas tiga proses yaitu pembangkitan kunci, enkripsi dan dekripsi. Karena algoritma ini termasuk algoritma asimetris maka pada proses pembangkitan kunci dibangkitkan dua kunci yaitu kunci publik (n, e) dan kunci rahasia (d) oleh penerima pesan. Langkah-langkah dalam proses pembangkitan kunci adalah sebagai berikut.

a. Pilih dua bilangan prima sebarang p_1 dan p_2.
b. Hitung $n = p_1.p_2$
c. Hitung $\varphi(n) = (p_1 - 1)(p_2 - 1)$
d. Pilih kunci publik e yang relatif prima terhadap $\varphi(n)$.
e. Bangkitkan kunci rahasia d yang memenuhi $e.d \equiv 1(\mathrm{mod}\ \varphi(n))$.

Proses ini menghasilkan kunci publik (n, e) yang dikirimkan kepada pengirim pesan dan kunci rahasia (d) yang tetap disimpan oleh penerima pesan.

Setelah proses pembangkitan kunci, pengirim pesan menggunakan kunci publik (n, e) yang ia peroleh untuk mengenkripsi *plaintext* m. Langkah-langkah dalam melakukan enkripsi pesan adalah sebagai berikut.

a. Ubah setiap karakter *plaintext* menjadi digit desimal ASCII.
b. Nyatakan *plaintext* m menjadi blok-blok $m_1, m_2, \dots, m_i$ sedemikian sehingga setiap blok merepresentasikan nilai selang $[0, n-1]$.
c. Setiap blok m_i dienkripsi menjadi blok *ciphertext* c_i dengan rumus berikut.

$$c_i \equiv {m_i}^e \pmod{n}$$

Pengirim pesan kemudian mengirimkan *ciphertext* c kepada penerima pesan. Penerima pesan melakukan dekripsi pesan dengan langkah-langkah sebagai berikut.

a. Nyatakan *ciphertext* c menjadi blok-blok $c_1, c_2, \dots, c_i$ sedemikian sehingga setiap blok merepresentasikan nilai selang $[0, n-1]$.
b. Setiap blok c_i didekripsi menjadi blok m_i dengan rumus berikut.

$$m_i \equiv {c_i}^d \pmod{n}$$

Keamanan algoritma RSA ini didasarkan pada masalah pemfaktoran bilangan bulat atau *Integer Factorization Problem* (IFP). Pada penelitian [7, 9] dijelaskan tentang banyaknya operasi dan durasi waktu untuk memfaktorkan n seperti pada Tabel 2.1. yang membutuhkan waktu hingga empat miliar tahun jika dipilih n yang panjangnya 200 digit.

Tabel 2.1. Durasi waktu memfaktorkan n

Digit n	Banyaknya Operasi	Waktu
50	$1{,}4 \times 10^{10}$	3,9 jam
75	$9{,}0 \times 10^{12}$	104 hari
100	$2{,}3 \times 10^{15}$	74 tahun
200	$1{,}2 \times 10^{23}$	$3{,}8 \times 10^{9}$ tahun
300	$1{,}5 \times 10^{29}$	$4{,}9 \times 10^{15}$ tahun
500	$1{,}3 \times 10^{39}$	$4{,}2 \times 10^{25}$ tahun

3. KRIPTOGRAFI ELGAMAL

Algoritma kriptografi ElGamal adalah algoritma yang dikembangkan pertama kali oleh Taher ElGamal. Algoritma ini didasarkan pada algoritma Diffie dan Hellman [3]. Pada algoritma ElGamal digunakan grup perkalian $\mathbb{Z}_m{}^*$ yaitu $\mathbb{Z}_m{}^* = \{\bar{a} \in \mathbb{Z}_m : \gcd(a, m) = 1\}$ dan elemen primitif $a \in \mathbb{Z}_m{}^*$ seperti dijelaskan pada [2, 4, 6]. Sama seperti algoritma RSA, algoritma ini terdiri atas tiga proses yaitu pembangkitan kunci, enkripsi dan dekripsi.

Pada proses pembangkitan kunci, dibangkitkan dua kunci yaitu kunci publik (q, y, a) dan kunci rahasia (x) oleh penerima pesan. Langkah-langkah dalam proses pembangkitan kunci adalah sebagai berikut.

a. Pilih sembarang grup perkalian $\mathbb{Z}_q{}^*$ dengan q adalah bilangan prima aman yaitu bilangan prima yang memenuhi $q = 2.s + 1$ dengan s adalah bilangan prima.
b. Pilih a adalah elemen primitif dari $\mathbb{Z}_q{}^*$ dan $x \in \{0, \dots, q-2\}$.
c. Hitung $y = a^x \bmod q$.

Setelah proses pembangkitan kunci, pengirim pesan menggunakan kunci publik (q, y, a) yang ia peroleh untuk mengenkripsi *plaintext* m. Langkah-langkah dalam melakukan enkripsi pesan adalah sebagai berikut.

a. Ubah setiap karakter *plaintext* menjadi digit desimal ASCII.
b. Nyatakan *plaintext* m menjadi blok-blok $m_1, m_2, \dots, m_i$ sedemikian sehingga setiap blok merepresentasikan nilai selang $[0, n-1]$.
c. Pilih bilangan acak $k_i \in \{0, \dots, q-2\}$.
d. Hitung $\gamma_i \equiv a^{k_i} (\bmod q)$ dan $\delta_i \equiv y^{k_i}.m_i \pmod q$ dan susun pasangan γ_i dan δ_i menjadi *ciphertext* c.

Pengirim pesan kemudian mengirimkan *ciphertext* c kepada penerima pesan. Penerima pesan melakukan dekripsi pesan dengan menggunakan rumus berikut.

$$m_i \equiv \delta_i.(\gamma_i{}^x)^{-1} (\bmod q).$$

Keamanan algoritma ini didasarkan pada masalah logaritma diskrit atau *Discrete Logarithm Problem* (DLP). Meier [5] menjelaskan bahwa algoritma ElGamal aman dikarenakan oleh susahnya menyelesaikan masalah logaritma diskrit, juga dikarenakan dipilihnya bilangan acak k pada saat proses enkripsi yang dirahasiakan.

4. PENGEMBANGAN KRIPTOGRAFI RSA DAN KRIPTOGRAFI ELGAMAL

4.1. Algoritma Kriptografi RSA yang Ditingkatkan

Proses penyandian pesan menggunakan algoritma RSA yang ditingkatkan adalah sebagai berikut.

a. Pembangkitan Kunci

Pada tahap pembangkitan kunci, dilakukan beberapa langkah berikut.

1) Pilih tiga bilangan prima $p_1, p_2,$ dan p_3.
2) Hitung $n = p_1.p_2.p_3$ (sebaiknya $p_i \neq p_j, i \neq j$ agar n tidak mudah difaktorkan).
3) Hitung $\varphi(n) = (p_1 - 1)(p_2 - 1)(p_3 - 1)$.
4) Pilih kunci publik e, yang relatif prima terhadap $\varphi(n)$.
5) Bangkitkan d yang memenuhi $e.d = 1 \; mod \; \varphi(n)$.

b. Enkripsi

Pada tahap enkripsi, dilakukan beberapa langkah berikut.

1) Nyatakan *plaintext* m menjadi blok-blok $m_1, m_2, ...$
2) Hitung *ciphertext* dengan rumus $c_i = {m_i}^e mod \; n$.

c. Dekripsi

Pada tahap dekripsi, *plaintext* dapat dicari dengan rumus $m_i = {c_i}^d mod \; n$.

Perbedaan algoritma RSA standar dengan algoritma RSA yang ditingkatkan terdapat pada proses pembangkitkan kunci. Penambahan satu bilangan prima pada proses pembangkitan kunci mengakibatkan bilangan n yang diperoleh pada proses pembangkitan kunci algoritma RSA yang ditingkatkan lebih besar dari bilangan n pada proses pembangkitan kunci algoritma RSA standar. Hal ini mengakibatkan durasi waktu proses enkripsi dan dekripsi menjadi lebih cepat. Berikut diberikan algoritma untuk menghitung $a \bmod n$.

Algoritma menghitung $a \bmod n$
Input : Bilangan bulat a dan n.
Output : Bilangan bulat hasil perhitungan $a \bmod n$
Langkah:

1. Untuk $a > n$, kerjakan:
   $a \leftarrow a - n$.
2. Ouput (a).

Tabel 4.1. Durasi waktu proses enkripsi dan dekripsi

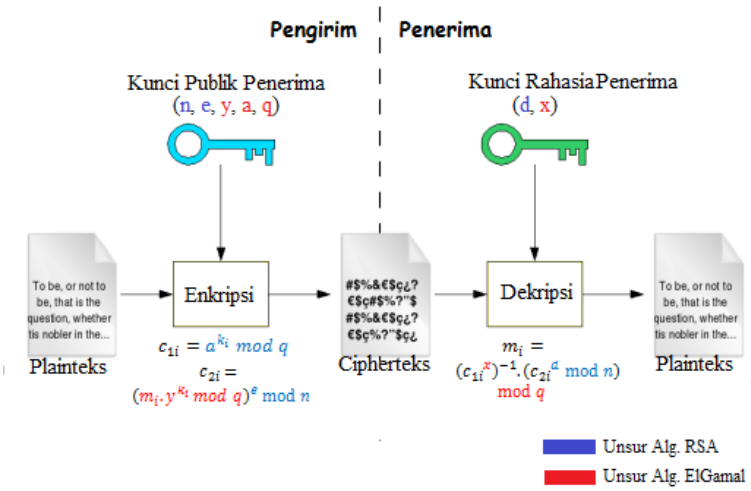
Algoritma RSA standar			**Algoritma RSA yang Ditingkatkan**		
p_i	**Digit n**	**Durasi Waktu**	p_i	**Digit n**	**Durasi Waktu**
43, 61	4	5,791 detik	43, 61, 7	5	5,554 detik
149, 307	5	8,926 detik	149, 307, 103	7	8,538 detik
1523, 1601	7	15,201 detik	1523, 1601, 1381	10	9,507 detik
7727, 8513	8	11,232 detik	7727, 8513, 6689	12	10,938 detik
Rata-rata waktu		**10,2875 detik**	**Rata-rata waktu**		**8,63425 detik**

Berdasarkan algoritma menghitung $a \bmod n$, semakin besar n maka proses perhitungan $a \bmod n$ akan lebih cepat. Pada Tabel 4.1. terdapat hasil pengujian durasi waktu proses enkripsi dan dekripsi pada algoritma RSA yang ditingkatkan dan algoritma RSA standar menggunakan Java Netbeans 8.2 menggunakan kalimat "Saya Mahasiswa Prodi Matematika Universitas Pendidikan Indonesia Angkatan 2013".

4.2. Kombinasi Algoritma RSA yang Ditingkatkan dan Algoritma ElGamal

Algoritma RSA yang ditingkatkan membutuhkan tiga bilangan prima $p_1, p_2,$ dan p_3. Untuk membentuk sistem kriptografi ElGamal, dibutuhkan bilangan prima q dan elemen primitif a pada grup perkalian $Z_q{}^*$. Oleh karena itu, algoritma kombinasi hasil penggabungan algoritma RSA yang ditingkatkan dan algoritma ElGamal membutuhkan tiga bilangan prima $p_1, p_2,$ dan p_3, bilangan prima q dan elemen primitif a pada grup perkalian $Z_q{}^*$ untuk membentuk sistem kriptografi.

Berikut adalah sistem pada algoritma kombinasi dari algoritma RSA yang ditingkatkan dan algoritma ElGamal.



Gambar 4.1. Sistem kombinasi algoritma RSA yang ditingkatkan dan algoritma ElGamal

a. Pembangkitan Kunci

1) Pilih grup perkalian Z_q^*, dengan q adalah bilangan prima aman, elemen primitif $a \in Z_q^*$, dan tiga bilangan prima p_1, p_2, p_3.
2) Hitung $n = p_1 . p_2 . p_3$.
3) Hitung $\varphi(n) = (p_1 - 1)(p_2 - 1)(p_3 - 1)$.
4) Pilih e yang relatif prima terhadap $\varphi(n)$.
5) Hitung d, di mana $d . e = 1 \bmod \varphi(n)$.
6) Pilih sebarang bilangan $x \in \{0,1, \dots, q-2\}$.
7) Hitung $y = a^x \bmod q$.

Kunci publik dari algoritma kombinasi ini adalah $n, e, y, a,$ dan q, sedangkan kunci rahasianya adalah d dan x.

Pada proses pembangkitan kunci ini digunakan Z_q^* dengan q adalah bilangan prima aman, yaitu bilangan prima yang memenuhi $q = 2.s + 1$ dengan s adalah prima, dengan tujuan untuk memudahkan dalam mengecek apakah suatu $a \in Z_q^*$ merupakan elemen primitif atau bukan.

b. Enkripsi dan Dekripsi

Pada proses ini, pesan dienkripsi menggunakan kunci publik (n, e, y, a, q) dan sebarang bilangan acak rahasia $k_i \in \{0,1, \dots, q-2\}$. Pertama, *plaintext* m dibagi menjadi blok-blok $m_0, m_1, \dots, m_i$, kemudian menentukan nilai *ciphertext* (c_{1i}, c_{2i}), yaitu:

$$c_{1i} = a^{k_i} \, mod \, q$$

dan

$$c_{2i} = \left(m_i . y^{k_i} \, mod \, q\right)^e \bmod n.$$

Pada proses dekripsi, pesan m dapat diperoleh dengan menghitung:

$$m_i = ({c_{1i}}^x)^{-1} . \left({c_{2i}}^d \bmod n\right) \bmod q.$$

Selanjutnya dilakukan pembuktian terhadap algoritma kombinasi tersebut, sebagai berikut:

${c_{2i}}^d \bmod n = \left(m_i . y^{k_i} \bmod q\right)^{e.d} \bmod n.$

Karena $e.d = 1 \bmod \left(\varphi(n)\right)$ maka $e.d = 1 + k.\varphi(n)$, sehingga

$$\begin{aligned} {c_{2i}}^d \bmod n &= \left(m_i . y^{k_i} \bmod q\right)^{1+k.\varphi(n)} \bmod n \\ &= \left(m_i . y^{k_i} \bmod q\right) . \left(m_i . y^{k_i} \bmod q\right)^{k.\varphi(n)} \bmod n \\ &= \left(m_i . y^{k_i} \bmod q\right) . \left(\left(m_i . y^{k_i} \bmod q\right)^{\varphi(n)}\right)^k \bmod n \end{aligned}$$

Berdasarkan Teorema Euler, $\left(m_i . y^{k_i} \, mod \, q\right)^{\varphi(n)} \equiv 1 \bmod n$ sehingga

$${c_{2i}}^d \bmod n = m_i . y^{k_i} \bmod q.$$

Oleh karena itu,

$$\begin{aligned} ({c_{1i}}^x)^{-1} . \left({c_{2i}}^d \bmod n\right) &= ({c_{1i}}^x)^{-1} . (m_i . y^{k_i}) \bmod q \\ &= ({c_{1i}}^x)^{-1} . \left(m_i . (a^x)^{k_i}\right) \bmod q \\ &= \left(\left(a^{k_i}\right)^x\right)^{-1} . \left(m_i . a^{k_i . x}\right) \bmod q \\ &= a^{-k_i . x} . a^{k_i . x} . m_i \bmod q \end{aligned}$$

Karena a adalah elemen primitif, maka diperoleh

$$({c_{1i}}^x)^{-1} . \left({c_{2i}}^d \bmod n\right) \bmod q = m_i \bmod q.$$

Terbukti bahwa akan diperoleh *plaintext* m dengan menggunakan persamaan yang dikonstruksi pada proses enkripsi dan dekripsi.

4.3. Penerapan

Misalkan Bob membangkitkan kunci sebagai berikut.

a. Bob memilih tiga bilangan prima $p_1 = 3, p_2 = 13$ dan $p_3 = 37$, grup perkalian ${Z_{563}}^*$, elemen primitif $6 \in {Z_{563}}^*$.
b. $n = p_1 . p_2 . p_3 = 3.13.37 = 1443.$
c. $\varphi(n) = (p_1 - 1)(p_2 - 1)(p_3 - 1) = (3 - 1)(13 - 1)(37 - 1) = 2.12.36 = 864.$
d. Bob memilih $e = 13.$

e. $13.d \equiv 1(\text{mod } 864)$, sehingga $d = 133$.

f. Bob memilih $x = 8$ sehingga $y \equiv a^x(\text{mod } q) \Rightarrow y \equiv 6^8(\text{mod } 563) \Rightarrow y = 187$.

Diperoleh kunci publik $(n = 1443, e = 13, y = 187, a = 6, q = 563)$ dan kunci rahasia $(d = 133, x = 8)$. Lalu kunci publik $(n = 1443, e = 13, y = 187, a = 6, q = 563)$ dikirimkan pada Alice.

Misalkan Alice akan mengirimkan *plaintext* "Buku#007" dan mengenkripsinya dengan menggunakan kunci publik $(n = 1443, e = 13, y = 187, a = 6, q = 563)$ yang ia dapat dari Bob. Perhatikan Tabel 4.2. untuk melihat proses enkripsi oleh Alice dengan menggunakan program aplikasi yang telah dikonstruksi menggunakan Java Netbeans 8.2.

Tabel 4.2. Proses enkripsi kombinasi algoritma RSA yang ditingkatkan dan algoritma ElGamal

Plaintext (m_i)	B	u	k	u	#	0	0	7
Desimal ASCII	66	117	107	117	35	48	48	55
k_i	45	86	123	0	34	19	219	4
$c_{1i} = a^{k_i} \, mod \, q$	102	399	320	1	445	96	89	170
$c_{2i} = (m_i . y^{k_i} \, mod \, q)^e \text{ mod } n$	1209	303	1224	117	314	177	174	921

Dari Tabel 4.2., diperoleh *ciphertext* $c = (c_{1i}, c_{2i})$ yang akan dikirimkan kepada Bob.

Tabel 4.3. Proses dekripsi kombinasi algoritma RSA yang ditingkatkan dan algoritma ElGamal

c_{1i}	102	399	320	1	445	96	89	170
c_{2i}	1209	303	1224	117	314	177	174	921
$m_i = ({c_{1i}}^x)^{-1} . ({c_{2i}}^d \text{ mod } n) \text{ mod } q$	66	117	107	117	35	48	48	55
Plaintext (m_i)	B	u	k	u	#	0	0	7

Setelah Bob menerima *ciphertext* dari Alice, ia mendekripsi *ciphertext* dengan menggunakan kunci rahasia $(d = 133, x = 8)$ dan kunci publik $(n = 1443, e = 13, y = 187, a = 6, q = 563)$ yang telah ia bangkitkan. Dari proses dekripsi seperti pada Tabel 4.3., Bob dapat mengetahui *plaintext* yang dikirimkan oleh Alice yaitu "Buku#007".

5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, diperoleh kesimpulan sebagai berikut.

a. Penambahan satu bilangan prima pada proses pembangkitan kunci algoritma RSA mengakibatkan durasi waktu proses enkripsi dan dekripsi menjadi lebih cepat.
b. Diperoleh algoritma kriptografi yang baru hasil penggabungan algoritma RSA yang ditingkatkan kecepatannya dan algoritma ElGamal yang memuat masalah pemfaktoran bilangan bulat dan masalah logaritma diskrit.

6. DAFTAR PUSTAKA

[1] Ariyus, D. (2008). *Pengantar ilmu kriptografi: teori, analisis dan implementasi.* Andi.

[2] Buchmann, J. A. (2000). *Introduction to cryptography.* Springer-Verlag New York, Inc.

[3] Diffie, W., & Hellman, M. E. (1976). New direction in cryptography. *IEEE Transaction on Information Theory*, 29-40.

[4] Fraleigh, J. B., & Katz, V. J. (2003). *A first course in abstract algebra.* Addison-Wesley.

[5] Meier, A. V. (2005). The ElGamal cryptosystem. 1-13.

[6] Menezes, A., Oorschot, P. v., & Vanstone, S. (1997). *Handbook of applied cryptography.* CRC Press, Inc.

[7] Milanov, E. (2009). The RSA algortihm. 1-11.

[8] Rivest, R. L., A. S., & L. A. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Massachusetts Institute of Technology, Cambridge*, 1-15.