Riset dan E-Jurnal Manajemen Informatika Komputer
Vol. 1 Nomor 1, Oktober 2016
e-ISSN : 2541-1330
p-ISSN : 2541-1322

KOMBINASI ALGORITMA RSA DAN TRANSPOSISI SEBAGAI PENGAMAN DATA

Herri Siagian
Universitas Sumatera Utara Medan
herry.it.2007@gmail.com

Abstract

Information technology is growing has made it as the needs of both individuals, groups or companies, the security and confidentiality of the data becomes very important, so that the information can not be easily misused by parties who are not interested, therefore to secure messaging techniques required data security with steganography or cryptography, in steganographic techniques of data is converted into bits and inserted into digital media. The cryptographic techniques, the original message (plaintext) was converted into an encrypted message (ciphertext), in which to conduct the necessary encryption and decryption key. Algorithms transposition and Rivest Shamir Adleman (RSA) is a technique that can be used to perform encryption and decryption, by doing a combination of these techniques is expected to increase data security, so that unauthorized parties can not easily determine the message undisclosed.

Keywords : transposition, RSA, cryptographic.

I. `Pendahuluan

Perkembangan tekonologi informasi dan komunikasi yang sangat cepat memberi pengaruh yang besar kepada kehidupan manusia, penggunakan internet yang sudah menjadi kebutuhan membuat membuat kita berpikir seberapa aman data kita ketika kita bertukar informasi dalam jaringan, maraknya kejahatan dalam dunia maya dengan teknik-teknik seperti penyadapan, interupsi, fabrikasi, dsb membuat pertukaran data antar pengguna internet menjadi tidak aman (Ajib, S & Rico, T, 2011), berbagai metode keamanan telah banyak dikembangkan yang bertuuan untuk melindungi dan menjaga kerahasiaan data dari pihak-pihak yang tidak berhak, salah satu teknik tersebut adalah kriptografi. Perkembangan teknologi yang pesat membuat algoritma kriptografi terus berkembang, jika dipandang dari kunci, algoritma kriptografi terbagi menjadi 2 yaitu simetri dan asimetri, dimana simetri menggunakan kunci tunggal dan aismetri menggunakan dua buah kunci, yaitu kunci public dan kunci private, penggabungan dari beberapa algoritma diharapkan bisa memperkuat pengamanan data, sehingga data-data yang dipertukarkan dalam jaringan menjadi lebih aman.

2. Landasan Teori

2.1 Algoritma RSA

RSA adalah salah satu teknik kriptografi yang menggunakan kunci asimetri, kekuatan dari RSA algoritma ini teletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi factor-faktor prima (Maureen, L.C, 2011), RSA diciptakan oleh Ron (R)ivest, (S)hamir, dan Leonard (A)dleman, algoritma ini memiliki besaran-besaran sebagai berikut:

1. p dan q bilangan prima (rahasia)
2. n = p . q (tidak rahasia)
3. (ϕ n) = (p – 1) . (q – 1) (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (cipherteks) (tidak rahasia)

pengirim maupun penerima harus mengetahui nilai n dan e, dan hanya penerima saja yang mengetahui nilai d, ini merupakan algoritma enkripsi kunci umum dengan kunci umum sebesar KU = {e,n} dan kunci khusus sebesar KR = {d,n}, agar algoritma ini bisa memenuhi syarat sebagai enkripsi kunci umum yang baik, maka harus memenuhi ketentuan-ketentuan sebagai berikut:

1. Kemungkinan menemukan nilai e, d, n sedemikian rupa sehingga M ed = M mod n untuk semua M < n
2. Relative mudah menghitung M e dan C d untuk nilai M < n

2.1.1 Pembangkit Kunci RSA

Untuk membangkitkan kunci RSA secara otomatis diperlukan suatu langkah-langkah untuk membangkitkan kunci tersebut, adapun langkah-langkah pembangkit kunci RSA adalah:

1. Memilih dua bilangan prima p dan q, bilangan ini harus cukup besar (minimal 100 digit)
2. Menghitung n = p . q, Bilangan n disebut parameter security (sebaliknya p $\neq$ q, sebab jika p = q maka n = p^2 sehingga dapat diperoleh dengan menarik akar pangkat dua dari n)
3. Menghitung ϕ (n) = (p-1)(q-1)

e-ISSN : 2541-1330
p-ISSN : 2541-1322

4. Memilih bilangan bulat e dengan algoritma Euclid yaitu gcd(ϕ(n),e) = 1; dimana 1 < e < ϕ(n)
5. Menghitung d dengan rumus $d = e^{-1} \bmod \phi(n)$
   Atau e . d ≡ 1 (mod ϕ (n)).
   Perhatikan bahwa e . d ≡ 1 (mod ϕ(n)) ekivalen dengan e . d = 1 + k ϕ(n), sehingga secara sederhana d dapat dihitung dengan: d = (1+k . ϕ(n)) / e
   1. Kunci umum (kunci publik) adalah KU = {e,n}
   2. Kunci pribadi (kunci privat) adalah KR = {d,n}.
   3. n tidak bersifat rahasia, sebab n diperlukan pada perhitungan enkripsi/dekripsi

2.1.2. Enkripsi

Seseorang (sebut A) mengenkripsi pesan M untuk seseorang (sebut B), maka yang harus dilakukan oleh orang pertama adalah:

1. Teks asli dengan syarat M < n
2. Ambil Kunci public B yang otentik (n,e)
3. Tampilkan pesan sebagai integer M dalam interval [0, n-1]
4. Teks rahasia didapat dari C = M e (mod n)
5. Kirim C ke B

2.1.3 Dekripsi

Untuk mendekripsi, B melakukan langkah-langkah berikut:

1. Gunakan kunci pribadi d untuk menghasilkan M
2. Teks rahasia adalah C
3. teks asli diapat dari M = C d (mod n)

2.2 Algoritma Transposisi

Algoritma tranposisi yang digunakan akan dimodifikasi, dimana algoritma ini memerlukan sebuah kunci untuk melakukan transposisi, kunci terdiri dari karakter-karakter yang terdapat pada ASCII, nantinya kunci ini yang akan membentuk urutan tranposisi. Adapun langkah-langkah algoritma transposisi ini adalah sebagai berikut:

1. Masukkan sebuah kalimat yang akan dibuat sebagai kunci.
2. Kemudian ekstrak kunci tersebut, sehingga tidak ada karakter yang sama.
3. Masukkan *plain teks*
4. Lakukan Transposisi yang menghasilkan *cipherteks*

2.2.1 Kunci Enkripsi

Misalkan kunci yang kita gunakan adalah "*magister usu*",

1. Lakukan ekstrak kunci menjadi "*magister u"*.
2. Ambil indeks dari kunci secara ascending
3. Gunakan pola indeks yang dihasilkan untuk Melakukan transposisi pada *plain teks*.

Index	0	1	2	3	4	5	6	7	8	9
Ekstrak	m	a	g	i	s	t	e	r		u
Pola	8	1	6	2	3	0	7	4	5	9

Jika kunci diurutkan secara ascending akan menghasilkan " 'spasi' aegimrstu", dimana index dari :

- 'spasi' = 8
- a = 1
- e = 6
- g = 2
- i = 3
- m = 0
- r = 7
- s = 4
- t = 5
- u = 9

dengan demikian pola enkripsi yang nantinya dihasikan adalah [8, 1, 6, 2, 3, 0, 7, 4, 5, 9]

2.2.2 Enkripsi

Untuk melakukan enkripsi, kita gunakan dari pola index yang telah dihasilkan pada saat pembentukan kunci, misalkan teks yang akan kita enkripsi adalah :

'*selamat belajar algoritma pemrograman*'

Pola Enkripsi	8	1	6	2	3	0	7	4	5	9
Plain teks	**s**	**e**	**l**	**a**	**m**	**a**	**t**		**b**	**e**
Index Asc	0	1	2	3	4	5	6	7	8	9
Hasil	a	e	a	m		b	l	t	s	e

Pola Enkripsi	8	1	6	2	3	0	7	4	5	9
Plain teks	l	a	j	a	r		a	l	g	o
Index Asc	0	1	2	3	4	5	6	7	8	9
Hasil	a	a	r	l	g	j	a	l	o	

e-ISSN : 2541-1330
p-ISSN : 2541-1322

Pola Enkripsi	8	1	6	2	3	0	7	4	5	9
Plain teks	**r**	**i**	**t**	**m**	**a**		**p**	**e**	**m**	**r**
Index Asc	0	1	2	3	4	5	6	7	8	9
hasil	i	m	a	e	m	t	p	r	r	o

Pola Enkripsi	8	1	6	2	3	0	7	4	5	9
Plain teks	**o**	**g**	**r**	**a**	**m**	**a**	**n**			
Index Asc	0	1	2	3	4	5	6	7	8	9
Hasil	o	g	r	a	m	a	n			

Cipher teks yang dihasilkan adalah : *'aeam bltse aarlgjalo imaemtprrograman*'

2.2.3 Kunci Dekripsi

Kunci dekripsi diperoleh dari kunci enkripsi dengan mencocokkan string key enkripsi dengan pola enkripsi kemudian ambil index key enkripsi per karakter berdasarkan pola enkripsi.

Index	0	1	2	3	4	5	6	7	8	9
Key enkripsi		a	e	g	i	m	r	s	t	u
Pola enkripsi	8	1	6	2	3	0	7	4	5	9
Pola dekripsi	5	1	3	4	7	8	2	6	0	9

Pola enkripsi dengan angka:

- 0 pada karakter m memiliki index 5
- 1 pada karakter a memiliki index 1
- 2 pada karakter g memiliki index 3
- 3 pada karakter i memiliki index 4
- 4 pada karakter s memiliki index 4
- 5 pada karakter t memiliki index 8
- 6 pada karakter e memiliki index 2
- 7 pada karakter r memiliki index 6
- 8 pada karakter '*spasi'* memiliki index 0
- 9 pada karakter u memiliki index 9

Jika kita lihat, maka urutan ascending pola enkripsi akan menghasilkan string '*magister u*' yaitu kunci itu sendiri dan pola dekripsi yang dihasilkan adalah: [5, 1, 3, 4, 7, 8, 2, 6, 0, 9]

2.2.4 Dekripsi

Untuk melakukan dekripsi kita gunakan pola dekripsi yang dihasilkan, *cipher teks* yang akan kita dekripsi adalah *'aeam bltse aarlgjalo imaemtprrograman'*

Pola dekripsi	5	1	3	4	7	8	2	6	0	9
Cihper teks	a	e	a	m		b	l	t	s	e
Index Asc	0	1	2	3	4	5	6	7	8	9
Hasil	s	e	l	a	m	a	t		b	e

Pola dekripsi	5	1	3	4	7	8	2	6	0	9
Cihper teks		a	a	r	l	g	j	a	l	o
Index Asc	0	1	2	3	4	5	6	7	8	9
Hasil	l	a	j	a	r		a	l	g	o

Pola dekripsi	5	1	3	4	7	8	2	6	0	9
Cihper teks		i	m	a	e	m	t	p	r	r
Index Asc	0	1	2	3	4	5	6	7	8	9
Hasil	r	i	t	m	a		p	e	m	r

Pola dekripsi	5	1	3	4	7	8	2	6	0	9
Cihper teks	o	g	r	a	a	m	a	n		
Index Asc	0	1	2	3	4	5	6	7	8	9
Hasil	o	g	r	a	a	m	a	n		

Ambil karakter pada pola enkripsi dan urutkan dengan angka:

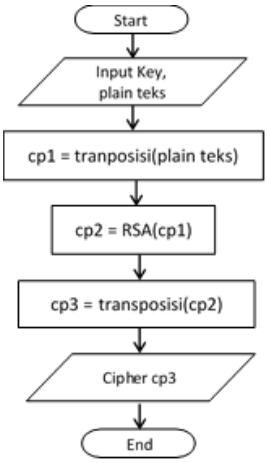
- 0 pada pola enkripsi akan menghasilkan = s
- 1 pada pola enkripsi akan menghasilkan = e
- 2 pada pola enkripsi akan menghasilkan = l
- 3 pada pola enkripsi akan menghasilkan = a
- 4 pada pola enkripsi akan menghasilkan = m
- 5 pada pola enkripsi akan menghasilkan = a
- 6 pada pola enkripsi akan menghasilkan = t
- 7 pada pola enkripsi akan menghasilkan = 'spasi
- 8 pada pola enkripsi akan menghasilkan = b
- 9 pada pola enkripsi akan menghasilkan = e

e-ISSN : 2541-1330
p-ISSN : 2541-1322

Dengan demikan akan menghasilkan *plain teks* '*selamat belajar algoritma pemrograman'*
Pada kalimat '*ograman'* tidak dilakukan pengurutan, hal ini dikarenakan kunci lebih panjang dari *cipher teks.*

3. Analisa dan Pembahasan

Kombinasi algoritma yang dilakukan adalah dengan menggabungkan algoritma Transposisi dan RSA dimana terbagi dalam beberapa langkah, seperti pada gambar beikut:



Gambar 1. Alur Kombinasi RSA Transposisi

1. Berikan sebuah kunci dan *plain* teks, dimana kunci ini nantinya akan digunakan sebagai pembentuk permutasi pada algoritma transposisi.
2. Lakukan enkripsi pada *plain* teks dengan Transposisi (ch1).
3. Berikan / Bangkitkan kunci publik dan private algoritma RSA.
4. Lakukan enkripsi cp1 dengan RSA (cp2).
5. Lakukan enkripsi cp2 dengan Tranposisi (cp3).
6. Dihasilkan *cipher* teks hasil kombinasi algoritma.

Proses perhitungan enkripsi kombinasi algoritma adalah sebagai berikut:

1. Misalkan, plainteks = '*belajar algoritma ok*'
   Kunci transposisi = '*teknik*'
   Kunci publik RSA = {7, 187}, kunci privat = {23, 187}
2. Lakukan ekstrak kunci dan hasilkan pola transposisi
   Ekstrak '*teknik*' menjadi '*tekni*'
   Pola transposisi = [1, 4, 2, 3, 0]
3. Lakukan transposisi pada plainteks
   Bagi plainteks sesuai panjang kunci = ['belaj', 'ar al', 'gorit', 'ma ok']
   Hasil transposisi = ['jblae', 'la ar', 'tgrio', 'km oa'] = 'jblaela artgriokm oa'
4. Lakukan enkripsi dengan kunci publik {7, 187}
5. Pecah hasil transposisi menjadi blok-blok m_i
   [106, 98, 108, 97, 101, 108, 97, 32, 97, 114, 116, 103, 114, 105, 111, 107, 109, 32, 111, 97]
   Hitung nilai enkripsinya:

$m_1 = 106^7 \ mod \ 187 = 149$	$m_{11} = 98^7 \ mod \ 187 = 74$
$m_2 = 106^7 \ mod \ 187 = 21$	$m_{12} = 98^7 \ mod \ 187 = 137$
$m_3 = 106^7 \ mod \ 187 = 48$	$m_{13} = 98^7 \ mod \ 187 = 126$
$m_4 = 106^7 \ mod \ 187 = 92$	$m_{14} = 98^7 \ mod \ 187 = 96$
$m_5 = 106^7 \ mod \ 187 = 84$	$m_{15} = 98^7 \ mod \ 187 = 155$
$m_6 = 106^7 \ mod \ 187 = 48$	$m_{16} = 98^7 \ mod \ 187 = 112$
$m_7 = 106^7 \ mod \ 187 = 92$	$m_{17} = 98^7 \ mod \ 187 = 131$
$m_8 = 106^7 \ mod \ 187 = 76$	$m_{18} = 98^7 \ mod \ 187 = 76$
$m_9 = 106^7 \ mod \ 187 = 92$	$m_{19} = 98^7 \ mod \ 187 = 155$
$m_{10} = 106^7 \ mod \ 187 = 126$	$m_{20} = 98^7 \ mod \ 187 = 92$

6. Hasil enkripsi RSA adalah '1492148928448927692126741371269615511213176155 92'
7. Lakukan enkripsi transposisi dari cipherteks yang telah dihasilkan dengan kunci yang sama pada saat melakukan transposisi pertama sekali. Sehingga mengasilkan cipherteks = '1142484489869792746112613951561173269251'

Dari hasil penggabungan algoritma akan dihasilkan cipherteks yang lebih kuat, hal ini didasarkan dari cipherteks yang dihasilkan menjadi lebih acak hal ini dikarenakan terdapat penambahan lapisan keamanan, sehingga *cryptanalis* perlu mememecahkan kunci transposisi terlebih dahulu.

4. Kesimpulan

Penerapan algoritma kombinasi ini lebih optimal pada teks yang tidak terlalu panjang, hal ini dapat dilihat dari waktu yang dihasilkan pada saat melakukan enkripsi dan dekripsi dimana semakin panjang teks yang akan dienkripsi ataupun didekripsi maka semakin lama waktu yang dibutuhkan, dengan demikian penerapan algoritma ini sangat cocok untuk mengamankan kunci dari algoritma kriptografi lainnya ataupun pada pesan-pesan yang singkat.

Semakin panjang kunci transposisi akan semakin menyulitkan *cryptanlis* pada saat melakukan dekripsi dari algoritma ini. Hal ini dikarenakan semakin banyak pola

e-ISSN : 2541-1330
p-ISSN : 2541-1322

enkripsi yang akan diuji jika *cryptanalis* melakukan serangan metode *bruteforce*.

Jumlah karakter hasil enkripsi yang semakin panjang akan semakin menyulitkan *cryptanalis* dalam melakukan dekripsi dari pesan yang telah enkripsi sebelumnya.

Daftar Pustaka

[1] Ariyus, D. 2008. *Pengantar Ilmu Kriptografi*. Andi Offset: Yogyakarta.
[2] Arifin, Zainal. 2009, Studi kasus penggunaan algoritma RSA sebagai algoritma kriptografi yang aman, *Informatika Mulawarman***4**(3):7-4
[3] Booth, Paul A. 1989. An Introduction to Human-Computer Interaction, Lawrence Erlbaum Associates Ltd: Inggris.
[4] Caoline Linda Maureen, 2011. "Perbandingan Algoritma Kriptografi Publik RSA, Rabin dan ElGamal". *Jurnal ITB*
[5] Childs, Lindsay N. 2000. A Concrete Introduction to Higher Algebra, Undergraduate Texts in Mathematics. Springer-Verlaag: New York.
[6] Kurniawan, Yusuf. 2004. Kriptografi keamanan internet dan jaringan komunikasi. Informatika Bandung
[7] Kromodimoeljo, S. 2010. Teori dan aplikasi kriptografi. SPK IT Consulting.
[8] Listiyono, Hersatoto. 2009. Implementasi algoritma kunci publik pada algoritma RSA. *Dinamika Informatika***1**(2):95-99.
[9] Menezes, A.J., Oorschot, P.V. & Vanstone, S. 1996. Handbook of Applied Cryptography. CRC Press: New York