ALGORITMA *DIGITAL SIGNATURE SOLIN* SEBAGAI PENGAMANAN JARINGAN TELEKOMUNIKASI

Sofu Risqi Y.S[1)], Lintang Kesumastuti[2)]
[1] Lembaga Sandi Negara [2] Badan Meteorologi Klimatologi dan Geofisika
E-mail : sofurizky@yahoo.com[1)], lintangkst@gmail.com[2)]

Abstrak

Tanda tangan merupakan salah satu penjamin keaslian data dari penanda tangan. Terdapat dua jenis tanda tangan yaitu tanda tangan manual dan digital. SOLIN *adalah algoritma tanda tangan digital (digital signature) yang dapat menjamin otentikasi dan integritas data.* SOLIN *dalam menjamin otentikasi memanfaatkan masalah pemfaktoran bilangan prima. Sedangkan dalam menjamin integritas datanya menggunakan algoritma fungsi hash SR-11. Dari hasil analisis yang didapatkan* SOLIN *sulit untuk dilakukan pemalsuan sehingga dapat dimanfaatkan untuk mengamankan jaringan telekomunikasi untuk menjamin keaslian data yang dikirimkan.*

Kata kunci: *SOLIN, Otentikasi, Integritas, SR-11*

Abstract

Signature is a guarantors of the data authenticity. There are two types of signatures that are manual and digital signatures. SOLIN *is a digital signature algorithm (digital signature) to ensure authentication and data integrity.* SOLIN *in ensuring authentication utilizing factoring primes problem. While in ensuring the integrity of the data using a hash function algorithm SR-11. From the analysis results obtained* SOLIN *counterfeiting difficult to do so can be used to secure telecommunications network to ensure the authenticity of data transmitted.*

Keywords: *SOLIN, Authentication, Integrity, SR-11*

1. PENDAHULUAN

Dalam melakukan komunikasi, suatu pesan atau data yang dikirim sangat perlu dijamin akan autentikasi kepemilikannya terutama bagi si penerima agar terjadi keyakinan bahwa sumber pesan yang diterima merupakan pesan yang asli. Kepemilikan pesan, data atau surat dapat diautentikasi dengan melihat tanda tangan yang tertera pada surat atau pesan tersebut. Tanda tangan tersebut mengartikan bahwa penandatangan setuju dan telah membaca surat atau pesan tersebut. Dengan kata lain, surat atau pesan tersebut telah berkorespondensi dengan identitas pengirimnya.

Selain tanda tangan yang konvensional (manual), dalam dunia kriptografi terdapat juga tanda tangan dalam bentuk *digital* atau biasa disebut *digital signature*. *Digital signature* digunakan untuk menanda tangani suatu pesan dalam bentuk *digital*. Tanda tangan tersebut berfungsi untuk autentikasi terhadap kepemilikan dari pesan *digital* tersebut.

Kriptografi adalah ilmu teknik matematika yang berelasi dengan aspek pada keamanan informasi sehingga diharapkan memenuhi *confidentiality, data integrity, entity authentication,* dan *data origin authentication* [1]. *Digital signature* dapat digunakan untuk menjamin *data integrity*, *entity authentication,* dan *data origin authentication*. Terdapat banyak skema *digital signature* yang ada dan dapat digunakan sebagai alat otentikasi. Pada tulisan ini penulis mencoba mengusulkan sebuah skema *digital signature* baru. Dimana kami beri nama skema SOLIN *signature* yang mana adalah hasil modifikasi dari skema RSA *digital signature*. SOLIN *signature* memiliki keamanan yang lebih baik dari RSA *digital signature* karena memiliki parameter tambahan guna mempersulit penyerang untuk memalsukan tanda tangan tersebut. Sebelum dilakukan tanda tangan data dilakukan proses *hashing* dengan algoritma *hash* SR-11untuk menjamin keutuhan data, kemudian dilakukan penanda tanganan pada data tersebut. Dengan adanya tanda tangan *digital* ini siapapun dapat membuktikan kepemilikan data yang telah ditanda tangani tanpa ada penyangkalan dari si-penanda tangan sehingga dapat dijadikan bukti pada persidangan. Tujuan dari tulisan ini yaitu untuk mengatasi masalah perkotaan yang menuntut

kecepatan dan keamanan dalam proses transaksi data, khususnya data dalam bentuk *digital*. Maka dengan penerapan matematika yaitu masalah faktorisasi bilangan prima dapat dijadikan solusi sebagai penjamin keaslian data yang dikirimkan guna meningkatkan aspek efisiensi waktu, biaya dan sumberdaya.

2. LANDASAN TEORI

Berikut dijelaskan landasan teori yang mendukung dalam penelitian ini:

2.1 Teori Bilangan

1. Faktor Persekutuan Terbesar (*Greatest Common Divisor*)[3]

Definisi 1

Untuk $a, b \in$ Z, *Greatest Common Divisor* dari a dan b didefinisikan sebagai positif integer d yang memenuhi:

a. $d|a$ dan $d|b$
b. Jika $c|a$ dan $c|b$ maka $c \leq d$

Integer d tersebut dapat dinyatakan dalam bentuk $d = au + bv$, dimana $u, v \in$ Z dinotasikan: $\gcd(a, b)$

Definisi 2

Integer a dan b dikatakan relatif prima jika $\gcd(a, b) = 1$. Berikut adalah sifat-sifat dasar gcd:

Untuk $a, b, k, m \epsilon$ Z:

a. a dan b relatif prima jika terdapat $u, v \epsilon$ Z sedemikian hingga $au + bv = 1$
b. $\gcd(a, b) = \gcd(b, a) = \gcd(|a|, |b|)$
c. $\gcd(ka, kb) = |k| \gcd(a, b)$
d. $\gcd(a, 0) = |a|$
e. $\gcd(a, 1) = 1$
f. $\gcd(a, b) = \gcd(a, b + ka)$, untuk semua $k \epsilon$ Z
g. Jika $\gcd(a, m) = \gcd(b, m) = 1$ maka $\gcd(ab, m) = 1$
h. Jika $\gcd(a, b) = 1$ maka $\gcd(a^k, b^l) = 1$, untuk semua $k, l \epsilon$ N

2. Aritmatika Modular

Aritmatika modular memiliki perhitungan yang secara logis akan lebih kompleks daripada perhitungan biasa, karena memerlukan pembagian oleh modulusnya. Namun hal ini oleh beberapa algoritma dijadikan suatu pegangan untuk mengurangi jumlah perulangan yang dibutuhkan. Selain itu hasil dari operasi matematik modular dibatasi oleh modulus bilangan. Sehingga kontrol terhadap panjang bilangan terutama bilangan berdigit besar akan lebih mudah diimplementasikan daripada aritmatika biasa.

Operasi-operasi pada aritmatika modular memiliki perbedaan dengan aritmatika biasa yakni dengan adanya modulus. Modulus adalah integer pembagi sehingga hasil-hasil yang didapat tidak akan lebih dari modulusnya. Bila dinotasikan sebagai berikut [4]:

$p \ mod \ n = q$ dimana $0 \leq q < n$, dan $p, q, n \in$ Z

Jika dirumuskan sebagai berikut :

$p = xn + q$ dimana x adalah hasil bagi bulat dan q adalah *remainder* (sisa)

Operasi-operasi modular memiliki sifat sebagai berikut:

1. $(a + b) mod \ n = \big((a \ mod \ n) + (b \ mod \ n)\big) mod \ n$
2. $(a - b) mod \ n = \big((a \ mod \ n) - (b \ mod \ n)\big)$
3. $(a.b) \ mod \ n = \big((a \ mod \ n) \ . (b \ mod \ n)\big) mod \ n$
4. $(a \ . (b + c) mod \ n = \big(\big((a \ . b) mod \ n\big) + \big((a \ . c) mod \ n\big)\big)$
5. $(\lfloor a/b \rfloor mod \ n \neq ([((a \ mod \ n))/ \ (b \ mod \ n) \ ]) mod \ n$

Berdasarkan sifat-sifat ini maka pengolahan bilangan pada aritmatika modular terutama sekali perhitungan perkalian dan pemangkatan dapat dilaksanakan dengan cara yang beragam tergantung dari hasil yang diinginkan, besarnya masukan dan waktu proses.

3. Kongruen [4]

Misal n integer positif. Jika x dan y adalah integer, maka x dikatakan kongruen terhadap y modulus n, ditulis:

$$x \equiv y \ (mod \ n)$$

Jika n membagi $(x - y)$ sedemikian sehingga $x - y = kn$ untuk integer k yang memenuhi. Integer n disebut modulus dari sifat kongruen.

Contoh:

i. $24 \equiv 9\ (mod\ 5)$ karena $24 - 9 = 3.5$
ii. $-11 \equiv 17\ (mod\ 7)$ karena $(-11) - 17 = (-4).7$

4. Pemangkatan Modular [3]

Untuk mengimplementasikan sistem sandi RSA, maka setidaknya fungsi enkripsi digunakan sebagai contoh .

$E\ (M) = M^e\ (mod\ n)$

Dimana e dan n adalah integer berdigit besar. Jika bernilai besar maka menghitung M^e secara langsung memerlukan memori pemprogramanan yang sangat besar. Sebenarnya permasalahan ini dapat diselesaikan dengan menggunakan metode biner, dengan mengubah e menjadi bit-bit yang mewakil. Oleh karena itu pemangkatan modular yang dijelaskan adalah pemangkatan modular dengan metode biner. Metode biner membaca bit-bit pangkat dari kiri ke kanan maupun dari kanan ke kiri. Pemangkatan dua dilakukan setiap langkah, kemudian dilakukan perkalian sesuai harga bit yang terbaca. Berikut akan dijelaskan metode biner dari kanan ke kiri.

Berikut ini adalah algoritma metode biner dari kanan ke kiri adalah sebagai berikut [3] :

a. Input: M, e, n
b. Proses :
   $if\ e_{k-1} = 1\ then\ C = M\ else\ then\ C = 1$
   $for\ i = k - 2\ to\ 0\ do$
   $\{\ C = C.C\ mod\ n$
   $if\ e_i\ then\ C = C.M\ mod\ N\}$
   $return\ (C)$
c. Output : $C = M^e\ mod\ N$

5. Invers Modular [3]

Sebuah integer y dikatakan invers dari $x\ mod\ n$, ditulis : $xy \equiv 1\ (mod\ n)$ atau dalam bentuk lain $xy = kn + 1$, untuk integer k. Operasi ini sangat penting dalam sistem sandi RSA. Invers modular sering dinotasikan dengan $x^{-1}\ mod\ n$ dan x akan memiliki $invers\ mod\ n$ apabila $\gcd(x, n) = 1$ atau saling relatif prima.

Contoh : $9 = 17^{-1}\ mod\ 19$ karena $9.17 = 153 = 1\ (mod\ 19)$

6. Teorema Fermat [3]

Jika p adalah bilangan prima dan a adalah positif integer yang bukan kelipatan dari p, maka :

$$a^{p-1} \equiv 1\ (mod\ p)$$

7. Fungsi Euler's Phi Totient $(\emptyset(n))$ [4]

Fungsi Euler $\emptyset$ menyatakan jumlah integer yang termasuk dalam himpunan $\{1, \dots, n-1\}$ yang relatif prima terhadap n, dinotasikan dengan : $\emptyset(n)$.

$\emptyset(1)$ didefinisikan sebagai 1. Untuk menghitung jumlah integer yang relatif prima terhadap n, faktorkanlah sedemikian sehingga memenuhi persamaan $n = p_1 p_2 \dots p_k$ (dimana $p_1 p_2 \dots p_k$ merupakan faktor prima dari n. Untuk menghitungnya gunakan rumus :

$\emptyset(n) = n(1 - {p_1}^{-1})(1 - {p_2}^{-1}) \dots (1 - {p_k}^{-1})$

Beberapa sifat-sifat Euler Totient yang penting bagi sistem sandi RSA:

a. Jika p adalah bilangan prima maka $\emptyset(p) = p - 1$, karena bilangan prima tidak memiliki pembagi kurang dari dirinya selain 1
b. Jika p dan q adalah bilangan prima, maka $\emptyset(pq) = \emptyset(p).\emptyset(q) = (p-1).(q-1)$
c. Jika p adalah bilangan prima dan $m > 0$, maka $\emptyset(p^m) = p^m - (p^{m-1})$
d. Jika $p$dan r adalah bilangan prima, maka $\emptyset(p^m r^n) = p^m r^n (p^{m-1})(r^{n-1})$

8. Teorema Euler [4]

Teorema ini menyatakan bahwa setiap a dan n yang saling relatif prima, maka :

$a^{\emptyset(n)} \equiv 1 \; mod \; n$

Pembuktian:

Jika n adalah bilangan prima rumus sudah terbukti, karena $\emptyset(n) = (n-1)$, jadi teorema Fermat terpenuhi. Tetapi kalau n bukanlah prima akan dibuktikan berikut ini. Misal R adalah himpunan dari $\emptyset(n)$:

$R = \{x_1 x_2, \dots, x_{\emptyset(n)}\}$

Lalu kalikan setiap elemen dengan $modulus \; n$:

$S = \{(ax_1 \; mod \; n), (ax_2 \; mod \; n), \dots, (ax_{\emptyset(n)} mod \; n)\}$

Himpunan S merupakan permutasi dari R, dengan alasan :

a. Karena a relatif prima terhadap n dan $X_i$relatif prima terhadap n, maka aX_ijuga relatif prima terhadap n yang relatif prima terhadap n.
b. Tidak ada elemen yang bernilai sama pada himpunan S. Karena jika $aX_i mod \; n = aX_j mod \; n$, maka $X_i = X_j$. Oleh karena itu :

$$\prod_{i=1}^{\emptyset(n)} (ax_i \; mod \; n) = \prod_{i=1}^{\emptyset(n)} x_i$$

$$\prod_{i=1}^{\emptyset(n)} (ax_i) \equiv \prod_{i=1}^{\emptyset(n)} x_i \; (mod \; n)$$

$$a^{\emptyset(n)} + \prod_{i=1}^{\emptyset(n)} x_i \equiv \prod_{i=1}^{\emptyset(n)} x_i \; (mod \; n)$$

$$a^{\emptyset(n)} \equiv 1 \; (mod \; n)(QED)$$

2.2 Konsep *Digtal Signature*

Digital signature adalah sebuah pesan berupa angka-angka yang bergantung pada beberapa *secret*(rahasia) yang hanya diketahui oleh *signer*(penandatangan pesan) dan sebagai tambahan, juga bergantung pada konten dari pesan yang ditandatangani. *Signature*(tanda tangan) harus bisa diverifikasi.

Digital signature memiliki beberapa kriteria keamanan informasi, yaitu: otentikasi, integritas data dan nir-penyangkalan(*non-repudiation*).Digital signature dibagi menjadi 2 tipe yaitu :

1. *Digital signature scheme* dengan *appendix*
   Tipe *digital signature* ini membutuhkan pesan asli sebagai *input* dalam proses verifikasi (untuk pesan sengan panjang berubah ubah)
2. *Digital signature scheme* dengan *recovery* pesan
   Tipe *digital signature* ini tidak memerlukan pesan asli (untuk pesan dengan panjang tetap). Pesan asli akan diperoleh dari *signature* itu sendiri (*message recovery*). Tipe ini memerlukan fungsi redudansi terhadap pesan sebelum memasuki proses *signature generation* (pembangkitan signature)

Tipe *digital signature* pesan dapat dikombinasikan dengan *appendix* dengan cara melakukan hash terhadap pesan sebelum masuk kedalam fungsi redudansi

A. Tipe serangan pada *signature scheme* :

1. Total breake
   Dapat menghitung *private key signer.*
2. Selective forgery
   Bisa membuat valid signature untuk pesan khusus.
3. Existensial forgery
   Bisa memalsukan paling tidak satu pesan.

B. Terdapat dua serangan dasar terhadap *public key digital signature sheme*

1. *Key-only attack*
   Penyerang hanya mengtahui *public key* pengirim
2. *Message attack*
   Penyerang mengetahui pesan, *message attack* dibagi menjadi 3 yaitu:
   a. *Known message attack* : penyerang mempunyai signature dari satu set pesan tapi tidak dipilih.
   b. *Choosen message attack* : penyerang mendapat signature yang valid dari daftar pilihan pesan
   c. *Addaptive choosen message attack* : penyerang dapan menggunakan mesin signer.

2.3 Algoritma *Digital Signature* RSA

Key generation :

1) *Generate* dua buah bilangan prima p dan q
2) Hitung $n = pq$ dan $\emptyset(n) = (p-1)(q-1)$
3) Pilih sebuah random *integer* $e, 1 < e < \emptyset$, dimana $\gcd(e, \emptyset(n)) = 1$
4) Hitung $e.d \equiv 1 \left(mod\ \emptyset(n)\right)$ dimana $1 < d < \emptyset(n)$
5) *Public key* (n, e); *private key* adalah d

Signature Generation :

1) Hitung $M = R(m)$ dimana $M < n$
2) Hitung $s = M^d\ mod\ n$

Verification Generation :

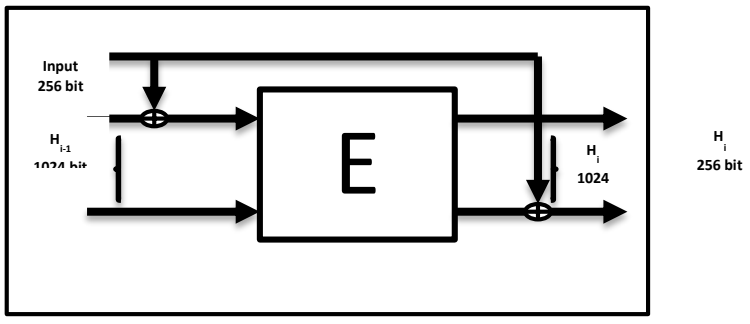
1) Diperoleh public key (n, e)
2) $M = s^e\ mod\ n$
3) Verifikasi M € M_R jika tidak tolak signing
4) *Recovery* $m = R^{-1}(M)$

Serangan pada digital signature RSA :

1. *Integer factorization*
   Penyerang dapat menentukan faktor dari nilai n, sehingga penyerang dapat menentukan $\emptyset(n)$ dengan menggunakan algoritma *extended Euclidean*
2. *Multiplicative property of RSA.*
   Serangan ini memanfaatkan komponen *multiplicative* pada RSA, terkadang sebagai *homomorphic property.*

2.4 Fungsi Hash SR-11

Algoritma *SR-11 Hash Function* merupakan algoritma *One Way Hash Function* yang dirancang oleh penulis dimana memiliki input pesan maksimal 2^{128} bit dengan output hash 256 bit. *SR-11 Hash Function* merupakan algoritma hash yang dengan konstruksi *Merkle Damgard* dengan satu kali proses tiap bloknya 256 bit. *SR-11 Hash Function* secara garis besar terdiri dari proses *round*, setiap *round* terdapat tiga buah fungsi yang berbeda, yaitu fungsi G, fungsi M dan fungsi Perkalian Matriks. Fungsi F, fungsi M dan fungsi Perkalian Matriks menggunakan komponen operasi penjumlahan, XOR, dan perkalian. Proses operasi didalam *SR-11 Hash Function* dengan IV sebanyak 1024 *bit* dan diproses dengan blok input 256 *bit.*



Gambar 1. Skema Fungsi Hash SR-11

3. METODOLOGI PENELITIAN

Berikut dijelaskan metodologi dalam melakukan penelitian ini:

3.1 Key *Generation*

1) *Generate* empat buah bilangan prima p_1 , q_1 dan p_2 , q_2
2) Hitung $n_1 = p_1 q_1$ dan $\emptyset(n_1) = (p_1 - 1)(q_1 - 1)$
3) Hitung $n_2 = p_2 q_2$ dan $\emptyset(n_2) = (p_2 - 1)(q_2 - 1)$
4) Pilih sebuah random *integer* e_1 , $1 < e_1 < \emptyset$, dimana $gcd\ (e_1 , \emptyset(n_1)) = 1$
5) Pilih sebuah random *integer* e_2 , $1 < e_2 < \emptyset$, dimana $gcd\ (e_2 , \emptyset(n_2)) = 1$
6) Hitung $e_1 . d_1 \equiv 1\ (mod\ \emptyset(n_1))$ dimana $1 < d_1 < \emptyset(n_1)$
7) Hitung $e_2 . d_2 \equiv 1\ (mod\ \emptyset(n_2))$ dimana $1 < d_2 < \emptyset(n_2)$
8) Syarat $n_1 < n_2$
9) *Public key* (n_1, n_2, e_1, e_2); *private key* adalah d_1, d_2

3.2 *Signature Generation*

1) Hitung $M = R(m)$ dimana $M < n$
2) Hitung $s = (M^{d_1}\ mod\ n_1)^{d_2}\ mod\ n_2$

3.3 *Verification Generation*

1) Diperoleh public key (n, e)
2) $M = (s^{e_2}\ mod\ n_2)^{e_1}\ mod\ n_1$
3) Verifikasi M € M_R jika tidak tolak signing
4) Recoveri $m = R^{-1}(M)$

4. PEMBAHASAN

Berikut ini berisi pembahasan tentang algoritma digital signature SOLIN:

4.1 Analisis Algoritma Digital Signature SOLIN

Kelebihan algoritma:

1) Algoritma SOLIN didesain berdasarkan modifikasi algoritma RSA *signature* diharapkan dapat memperbesar usaha kriptanalis dalam menemukan bilangan prima p dan q menggunakan *fermat factorization* karena harus menemukan dua nilai p dan q.
2) Kriptanalis harus menemukan nilai p_1 , p_2 terlebih dahulu kemudian q_1 , q_2 menggunakan *fermat factorization* dalam pencarian nilai p dan q. Terlebih lagi jika jarak antara nilai p dan q terpaut jauh, maka usaha yang dilakukan menggunakan *fermat factorization* akan semakin besar.

Kekurangan Algoritma:

1) Kekurangan ada algoritma SOLIN yaitu *signer* harus menentukan nilai p_1 , p_2 dan q_1 , q_2 .
2) Proses komputasi akan semakin besar karena menambahkan parameter p_1 , p_2 dan q_1 , q_2 .

4.2 Pembuktian

$$M = (S_2{}^{e_2}\ mod\ n_2)^{e_1}\ mod\ n_1$$

$$= ((S_1{}^{d_2})^{e_2}\ mod\ n_2)^{e_1}\ mod\ n_1$$

$$= (S_1)^{e_1}\ mod\ n_1$$

$$= (M^{d_1})^{e_1}\ mod\ n1$$

$$= M$$

5. KESIMPULAN

Berdasarkan data hasil penelitian, eksperimen dan analisis data, dapat ditarik simpulan Algoritma *digital signature* SOLIN merupakan algoritma modifikasi dari RSA (Rivest Shamir Adleman) *signature* yang mampu menjamin *data integrity*, *entity authentication*, dan *data origin authentication*. SOLIN dapat mempersulit kriptanalis dalam melakukan serangan *fermat factorization* untuk mendapatkan pemalsuannya atau nilai yang lain, sehingga data yang di kirimkan terjamin keasliannya. oleh karena itu SOLIN dapat dimanfaatkan untuk mengamankan jaringan telekomunikasi untuk menjamin keaslian data yang dikirimkan tanpa adanya penyangkalan dari pemilik data.

6. DAFTAR PUSTAKA

[1]. Menezes, Alfred J., Paul C. van Oorschot, Scott A. Vanstone. 1997. *Handbook of Applied Cryptography*. Boca Raton: CRC Press LLC.

[2]. Stalling, William.2005. *Cryptography and Network Security 4th Edition*, *Prentice Hall.*

[3]. Wagstaff, Samuel S. *Cryptanalisis of Number Theoreric Ciphers. Boca Raton*: CRC Press Company

[4]. Burton, D.M. 2005. *Elementary Number Theory, 6th ed*. McGraw-Hill Higher Education.

[5]. Bartkewitz, Timo. 2009. *Building Hash Functions from Block Ciphers, Their Security and Implementation Properties.* Ruhr-University Bochum.

[6]. Stinson, Douglas R. 2002. *Cryptography Theory and Practice, third edition.* Chapman & Hall/CRC.

[7]. Danda, M.K. Reddy. 2007. *Design and Analysis of Hash Functions.* Thesis in Network security and Cryptography, Victoria University.

[8]. Stamp, Mark. Richard M. Low. 2007. *Applied Cryptanalysis Breaking Ciphers in the Real World*. John Wiley & Sons, Inc.. Hoboken. New Jersey.

[9]. Yan, S.Y. 2002. *Number Theory for Computing, 2nd ed*. New York: Springer – Verlag Berlin Heidelberg.

[10]. Rosen, K.H. 1993. *Elementary Number Theory and Its Applications, 3rd ed*. Reading, MA: Addison-Wesley.

<halaman ini sengaja dikosongkan>