Analisis Implementasi Algoritma Kunci Publik pada Tanda Tangan Digital

Muhammad Luthfi 13507129

Program Studi Teknik Informatika,
Sekolah Teknik Elektro Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40135

email: luthfi@comlabs.itb.ac.id

Abstrak

Tanda tangan digital merupakan suatu bentuk otentifikasi dari sebuah file atau pesan yang berupa penanda tertentu yang hanya diketahui oleh pemberi tanda tangan tersebut. Tanda tangan digital diimplementasikan untuk dapat memverifikasi pengirim dari file atau pesan yang diterima seseorang. Hal ini dilakukan untuk mencegah pemalsuan otoritas dari suatu file atau pesan tersebut. Dengan tanda tangan digital, verifikasi dapat dilakukan tanpa harus mengetahui kunci private dari pengirim pesan.

Saat ini, tanda tangan digital banyak diaplikasikan dalam keamanan informasi. Termasuk diantaranya otentifikasi pengirim, kepercayaan pada file atau pesan yang dikirim, serta pencegahan dari penyangkalan pengiriman file atau pesan. Selain itu, tanda tangan digital juga dapat dikombinasikan dengan fungsi hash untuk dapat lebih menjamin integritas file atau pesan yang dikirim. Salah satu jenis tanda tangan digital yang paling banyak digunakan adalah sertifikat kunci publik pada jaringan internet.

Tujuan utama dari penerapan algoritma kunci publik sendiri adalah untuk menyediakan keleluasaan pribadi dengan tetap menjamin kerahasiaan dalam menyampaikan pesan secara rahasia atau terenkripsi. Dengan adanya pasangan kunci publik dan kunci private yang unik, maka pengirim dan integritas dari pesan yang dikirim dapat terjaga dengan baik. Hal ini sangat sesuai dengan kebutuhan akan otentifikasi pesan yang diterapkan pada tanda tangan digital.

Kata Kunci: *digital signature, public key algorithm*

1. PENDAHULUAN

Kriptografi dengan menggunakan proses enkripsi dan dekripsi data merupakan salah satu cara yang ampuh untuk digunakan dalam menjaga kerahasiaan data. Namun, aspek keamanan lain seperti otentifikasi, integritas atau keaslian data serta aspek anti-penyangkalan (nonrepudation) tidak dapat diselesaikan hanya dengan proses enkripsi dan dekripsi saja. Untuk dapat memenuhi hal tersebut, digunakanlah tanda tangan digital.

1.1 Pengertian Tanda Tangan Digital

Tanda tangan digital atau Digital Signature merupakan suatu tanda tangan (penanda) yang dibubuhkan pada data digital. Tanda tangan digital bukan merupakan hasil scan atau input tanda tangan melalui interface tertentu. Tanda tangan digital adalah suatu nilai kriptografis yang bergantung pada isi data itu sendiri serta kunci yang digunakan untuk membangkitkan nilai kriptografisnya. Sehingga nilai setiap tanda tangan digital dapat selalu berbeda tergantung data yang ditanda tangani.

Tanda tangan digital yang dibubuhkan kedalam suatu data dapat memvalidasi darimana data tersebut berasal. Tanda tangan ini memberi rasa aman kepada penerima data karena ia dapat mengetahui siapa yang mengirim data tersebut. Tanda tangan yang valid saat diotentifikasi ulang juga menjamin bahwa data yang dikirim tidak mengalami perubahan atau modifikasi selama proses pengiriman.

1.2 Penggunaan Tanda Tangan Digital

Sejak dahulu, tanda tangan telah digunakan sebagai media otentifikasi dokumen cetak. Tanda tangan yang dibubuhkan pada dokumen tersebut menunjukkan persetujuan dari orang yang memberi tanda tangan. Kini tanda tangan tersebut berkembang menjadi tanda tangan digital yang memiliki kegunaan yang jauh lebih banyak. Tidak saja untuk sekedar menandatangani dokumen-dokumen digital, tetap juga berbagai bentuk file dan keperluan lain seperti distribusi perangkat lunak, dan transaksi finansial perbankan.

2. TANDA TANGAN DIGITAL

Tanda tangan digital menerapkan teori kriptografi asimetrik. Kriptografi asimetrik merupakakan jenis kriptografi yang menggunakan kunci yang berbeda untuk mengenkripsi dan mendekripsi pesan. Kriptografi asimetri ini dimungkinkan setelah dikenalkannya kon-

sep penggunaan kunci publik dan kunci privat pada kriptografi. Dengan adanya sepasang kunci publik dan kunci privat ini, pengirim dapat membubuhi data yang dikirimkannya dengan tanda tangan yang telah dienkripsi. Tanda tangan digital yang telah dienkripsi ini lebih memberikan rasa aman dan kepercayaan bagi pihak penerima.

Tanda tangan digital secara umum terdiri dari tiga bagian (algoritma):

- *Public Key Generator*
  Algoritma untuk menggenerate sepasang kunci publik dan kunci privat. Kunci publik merupakan kunci yang digunakan untuk mengekripsi pesan atau dalam hal ini tanda tangan yang dibubuhkan, sedangkan kunci privat digunakan untuk mendekripsi ulang tanda tangan yang telah dienkripsi. Dengan cara ini, dapat diketahui darimana pesan itu berasal.

- *Hash Function*
  Algoritma untuk membuat tanda tangan digital atau Signing Algorithm. Fungsi hash akan menghasilkan nilai tertentu yang unik berdasarkan data yang digunakan sebagai masukan fungsi hash. Nilai hash ini bisa dijadikan tanda tangan bersama dengan proses enkripsi yang dilakukan. Sehingga tanda tangan yang dihasilkan pun benar-benar merepresentasikan pesan yang dikirim.

- *Verification Function*
  Algoritma untuk memverifikasi tanda tangan yang dibubuhkan. Bagian ini sebenarnya sama dengan bagian untuk mengekripsi tanda tangan. Karena dengan menggunakan pasangan kunci publik dan privat, seharusnya proses enkripsi dan dekripsi bisa dilakukan dengan algoritma yang sama dengan tambahan fungsi hash yang sama pula.

Suatu tanda tangan digital yang digenerate dari suatu pesan dan dienkripsi menggunakan kunci tertentu harus dapat digunakan untuk membuktikan keaslian pesan. Artinya fungsi hash dan proses enkripsi yang diterapkan harus membuat tanda tangan tersebut unik dan hanya bergantung pada data dan kunci masukan. Selain itu, kunci yang digunakan untuk proses enkripsi dan dekripsi harus merupakan kunci yang kuat yang tidak mudah didapatkan oleh pihak ketiga yang tidak memiliki kunci private pengirim.

2.1 Sejarah Tanda Tangan Digital

Catatan mengenai penggunaan tanda tangan digital mulai dibuat sejak Whitfield Diffie dan Martin Hellman pertama kali mendeskripsikan skema tanda tangan digital di tahun 1976. Walaupun pada saat itu keduanya baru sekedar mengusulkan bahwa penggunaan tanda tangan digital dapat dilakukan. Tidak lama setelah Diffie dan Hellman mengusulkan gagasan ini, Ronald Rivest, Adi Shamir, dan Len Adleman menemukan algoritma RSA yang dapat diterapkan pada skema awal tanda tangan digital yang diusulkan oleh Deffie dan Hellman tersebut.

Skema tanda tangan digital awal banyak menggunakan konsep yang sama, yaitu dengan memanfaatkan permutation. Permutation yang diterapkan pada algoritma RSA pada skema ini merupakan salah satu jenis permutasi yang mudah dilakukan namun sulit untuk dipecahkan. Parameter yang digunakan untuk prose enkripsi pada permutasi inilah yang menjadi kunci publik. Namun, bila parameter lain menjadi kunci privat diketahui, permutasi ini menjadi sangat mudah untuk dipecahkan.

Skema awal tanda tangan digital ini diketahui tidak terlalu aman. Sehingga kemudian diusulkan penambahan fungsi hash pada algoritma tanda tangan digital tersebut sehingga lebih tahan terhadap serangan yang diberikan. Selain tanda tangan digital yang diusulkan oleh Deffie dan Hellman, pada saat itu juga berkembang jenis tanda tangan digital lainnya seperti Lamport signatures, Merkle signatures dan Rabin signatures.

Fungsi hash yang ditambahkan pada algoritma tanda tangan digital tidak dilakukan pada keseleuruh pesan yang ditandatangani. Beberapa hal yang menjadi alasannya adalah efisiensi, kompabilitas dan integritas fungsi hash tersebut dengan skema tanda tangan digital. Dengan melakukan hash hanya pada tanda tangan , proses yang diperlukan untuk menandatangani pesan akan semakin cepat. Selain itu pesan yang ditandatangani tetap terjaga integritasnya.

2.2 Tingkat Keamanan pada Tanda Tangan Digital

Setelah diketahui berbagai kemungkinan serangan yang ditujukan pada tanda tangan digital, Shafi Goldwasser, Silvio Micali dan Ronal Rivest mencoba menetapkan tingkat keamanan yang perlu dipenuhi suatu algoritma tanda tangan digital. Mereka mendeskripsikan hirarki dari serangan yang mungkin ditujukan pada skema tanda tangan digital. Mereka juga memperlihatkan contoh skema tanda tangan digital yang terbukti dapat menanggulangi beberapa serangan yang dilancarkan.

Beberapa tipe serangan pada tanda tangan digital[2]:

- *Key only attack*
  Tipe serangan dimana penyerang hanya mengetahui kunci publik yang digunakan untuk proses enkripsi tanda tanga digital.

- *Known message attack*
  Tipe serangan dimana penyerang mengetahui beberapa tanda tangan valid yang dibubuhkan pada

pesan lain selain pesan yang sebenarnya menjadi tujuan serangan.

- *Adaptive chosen message attack*
  Tipe serangan dimana penyerang mempelajari algoritma tanda tangan digital melalui percobaan dengan membuat tanda tangan digital dari beberapa pesan yang dipilihnya.

Berdasarkan jenis-jenis serangan yang disebutkan diatas, Goldwasser, Micali dan Rivest juga menyebutkan hirarki dari hasil serangan yang dilakukan.

- *Total break*
  Total break merupakan kemungkian hasil serangan yang paling berbahaya. Total break adalah keadaan dimana penyerang dapat menemukan kunci yang diperlukan untuk mengenkripsi dan mendekripsikan tandan tangan digital.

- *Universal forgery*
  Universal forgery merupakan hasil penyerangan dimana pihak penyerang mampu memalsukan tanda tangan digital dari pesan manapun.

- *Selective forgery*
  Selective forgery serupa dengan universal forgery namun dalam lingkup yang lebih sempit. Penyerang hanya dapat memalsukan tanda tangan dari pesan yang diketahui olehnya.

- *Existential forgery*
  Existential forgery merupakan keadaan dimana penyerang hanya dapat mengira-ngira pasangan pesan dan tanda tangan yang validnya.

Jenis serangan yang dikemukakan diatas kebanyakan memiliki tujuan untuk melakukan pemalsuan tanda tangan. Hal ini serupa dengan banyaknya percobaan pemalsuan tanda tangan pada dokumen teks. Diharapkan dengan semakin berkembangnya teknologi kriptografi, bisa didapatkan cara yang efektif untuk mencegah terjadinya hal-hal seperti ini.

3. KRIPTOGRAFI KUNCI PUBLIK

Kriptografi kunci publik dikenal juga dengan istilah kriptografi asimetrik. Pada suatu kriptografi kunci publik, setiap kunci publik e akan memiliki suatu pasangan kunci privat e yang berkoresponden satu sama lain. Nilai kunci privat ini haruslah tidak mudah dipecahkan walaupun nilai e diberikan secara umum. Secara matematis, menghitung nilai d dari nilai e yang ada seharusnya sangat sulit dilakukan.

Kunci publik pada kriptografi kunci publik mendefinisikan parameter pada fungsi enkripsi E_e, sedangkan kunci privat mendefinisikan parameter untuk fungsi dekripsi D_d. Suatu pesan yang akan dikirim terlebih dahulu dienkripsi dengn menggunakan kunci publik penerima untuk mendapatkan cipherteks $c = E_e (m)$, dengan m adalah pesan yang akan dikirim. Untuk mendekripsinya digunakan kunci private penerima sehingga pesan $m = D_d (c)$ dapat diperoleh.

Kunci publik tidak perlu dirahasiakan keberadaannya. Bahkan kunci ini dapat disebarluaskan pada khalayak luas. Hanya saja, untuk keperluan enkripsi-dekripsi seorang pengirim harus mengetahui kunci publik penerima yang benar. Keungtungan utama dari sistem ini adalah kemudahan yang didapatkan untuk proses pengiriman pesan. Karena akan lebih mudah untuk menggunakan kunci publik yang diketahui secara umum daripada harus mengirimkan kunci untuk proses enkripsi-dekripsi seperti pada kriptorafi simetri.

Penggunaan kriptografi kunci publik adalah untuk menjaga privasi dan kerahasiaan dalam berkirim pesan secara rahasia. Hanya saja proses ini masih belum sempurna. Proses enkripsi yang dilakukan dengan menggunkan kunci publik yang telah disebarluaskan secara umum, menyebabkan kunci ini tidak dapat dipergunakan untuk menunjukkan darimana pesan tersebut berasal ataupun jaminan integritas dari pesan yang dikirim.

3.1 Kriptografi Kunci Publik pada Tanda Tangan Digital

Proses penggunaan skema kriptografi kunci publik untuk proses enkripsi-dekripsi pesan secara umum jauh lebih lambat dari kriptografi kunci simetri. Hal ini disebabkan kriptografi kunci publik melibatkan perhitungan angka-angka besar (big integer) baik pada proses enkripsi maupun dekripsinya. Karena alasan inilah, kriptografi kunci publik lebih sering digunakan untuk pengiriman kunci simetri dari pesan yang sebenarnya dikirim, tidak untuk mengenkripsi pesan itu sendiri.

Hal ini juga lah yang mendasari penggunaan kriptografi kunci publik pada tanda tangan digital. Message digest dari pesan yang ditanda tangani digenerate terlebih dahulu dengan menggunakan suatu fungsi hash tertentu. Namun, berbeda dengan skema kriptografi kunci publik pada umumnya. Kunci yang digunakan untuk mengekripsi message digest merupakan kunci privat pengirim. Ini dilakukan untuk sekaligus mendapatkan otentifikasi dari siapa pesan tersebut dikirim.

3.2 Algoritma Kriptografi Kunci Publik

Banyak algoritma kunci publik lain yang dapat dimanfaatkan pada tanda tangan digital. Beberapa diantaranya adalah RSA, Rabin Algorithm dan ElGamal. Terdapat pula algoritma lain seperti McEliece, Merkle-Hellman Knapsack, Chor-Rivest Knapsack, Goldwasser-Micali Probabilistic, serta Blum-Goldwasser Probabilistic. Setiap algoritma memiliki kelebihannya

masing-masing. Setiap kelebihan dari algoritma-algoritma ini berasal dari sulitnya komputasi yang dilakukan untuk mendapatkan parameter kuncinya.

Algoritma Kunci Publik	Keunggulan Komputasi
RSA	• integer factorization problem
ElGamal	• discrete logarithm problem • Deffie-Hellman problem
Rabin Algorithm	• integer factorization problem • square roots modulo composite n
McEliece	• linear code decoding problem
Merkle-Hellman	• knapsack subset sum problem
Chor-Rivest	• subset sum problem
Blum-Goldwasser	• probabilistic integer factorization problem • Rabin problem

Berikut akan dijelaskan beberapa algoritma kunci publik yang sering dipergunakan.

3.2.1 RSA

RSA merupakan algoritma kunci publik yang paling banyak digunakan. Nama algoritma kunci publik ini berasal dari nama penemunya, yaitu R.Rivest, A. Shamir, dan L. Adleman. RSA dapat diandalkan untuk menjamin kerahasiaan. Kekuatan dari algoritma RSA bersumber pada kesulitan untuk memfaktorisasi parameter-parameter kuncinya. Kesulitan pemfaktoran pada RSA berada pada permasalahan pemfaktoran integer.

Algoritma RSA:

1. Generate two large random (and distinct) primes p and q, each roughly the same size.
2. Compute $n = pq$ and $\varphi = (p - 1)(q - 1)$.
3. Select a random integer e, $1 < e < \varphi$, such that $\gcd(e, \varphi) = 1$.
4. Compute the unique integer d, $1 < d < \varphi$, such that $ed \equiv 1 \pmod{\varphi}$.
5. Public key is (n, e); Private key is d.

Proses enkripsi-dekripsi (B mengirim pesan m ke A)

1. *Encryption.* B should do the following:
(a) Obtain A's authentic public key (n, e).
(b) Represent the message as an integer m in the interval $[0, n - 1]$.
(c) Compute $c = m^e \bmod n$
(d) Send the ciphertext c to A.

2. *Decryption.* To recover plaintext m from c, A should do the following:
(a) Use the private key d to recover $m = c^d \bmod n$.

Beberapa serangan yang dikenal pada RSA antara lain *common modulus attack* dan *cyling attack*. *Common modulus attack* dapat dilakukan jika nilai e dan d diketahui serta ada pesan yang berkoresponden dengan kunci tersebut. Nilai e dan d ini biasanya hanya beredar di jaringan yang aman. Dengan diketahuinya nilai e dan d sebagai fakor dari modulus n, penyeranga dapat yang berasal dari jaringan lain dapat dengan mudah mendapatkan hasil dekripsi pesan.

3.2.2 ElGamal

Algoritma kriptografi kunci publik ElGamal berasal dari pengembangan Deffie-Hellman problem. ElGamal merupakan salah satu algoritma kriptografi yang pada proses enkripsinya menggunakan randomisasi. Selain ElGamal, terdapat pula Generalized ElGamal yang mengalami beberapa perubahan dengan penambahan konsep multiplicative group.

Dasar Algoritma ElGamal:

1. Generate a large random prime p and a generator α of the multiplicative group Z_p^* of the integers modulo p.
2. Select a random integer a, $1 \le a \le p - 2$, and compute $\alpha^a \bmod p$.
3. Public key is (p, α, α^a); Private key is a.

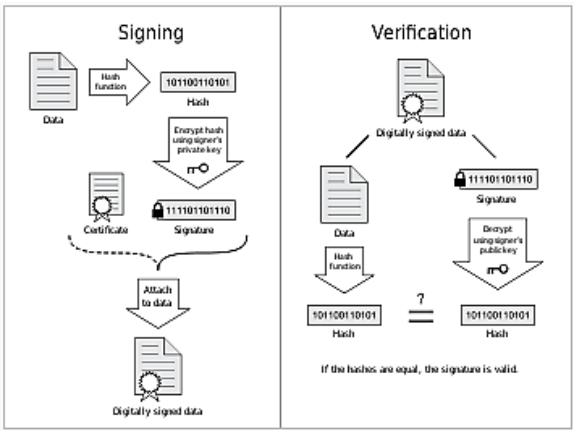
Proses enkripsi-dekripsi (B mengirim pesan m ke A)

1. *Encryption.* B should do the following:
(a) Obtain A's authentic public key (p, α, α^a).
(b) Represent the message as an integer m in the range $\{0, 1, \ldots, p - 1\}$.
(c) Select a random integer k, $1 \le k \le p - 2$.
(d) Compute $\gamma = \alpha^k \bmod p$ and $\delta = m\,(\alpha^a)^k \bmod p$.
(e) Send the ciphertext $c = (\gamma, \delta)$ to A.

2. *Decryption.* To recover plaintext m from c, A should do the following:
(a) Use the private key a to compute $\gamma^{p-1-a} \bmod p$ (note: $\gamma^{p-1-a} = \gamma^{-a} = \alpha^{-ak}$).
(b) Recover m by computing $(\gamma^{-a})\delta \bmod p$.

4. ALGORITMA TANDA TANGAN DIGITAL

Sebenarnya telah dikenal suatu standar internasional untuk algoritma yang digunakan pada tanda tangan digital. Standar tersebut adalah Digital Signature Algorithm (DSA) yang diumumkan oleh National Institute of Standards and Technology pada bulan Agustus 1991 untuk diterpakan pada Digital Signature Standard (DSS). DSA sendiri merupakan algoritma yang dibuat oleh David W. Kravitz. DSA terdiri dari beberapa tahap pengerjaan dimulai dari mengenerate kunci, penandatanganan dan proses verifikasi.



Untuk menggenerate kunci pada DSA, terlebih dahulu dipilih fungsi hash H yang akan digunakan. Pada DSS, fungsi hash yang digunakan ialah SHA-1, namun saat ini SHA-2 yang lebih kuat dari SHA-1 juga mulai dipergunakan pada DSS. Selain itu, perlu ditetapkan juga terlebih dahulu panjang kunci yang diinginkan. Panjang kunci ini mencerminkan kekuatan algoritma yang dibuat. DSS menggunakan kunci kelipatan 64 antara 512 hingga 1024 bits.

Berikut merupakan potongan kode yang digunakan untuk menggenerate parameter kunci untuk algoritma RSA yang dipergunakan pada Digital Signature Standard:

```
public void generate(int bitNums)
{
   BigInteger one = new BigInteger("1");
   BigInteger two = new BigInteger("2");
   java.util.Random r =
      new java.util.Random();

   //p dan q merupakan bilangan prima
      berbeda sebesar bitNums bit
   p = new BigInteger(bitNums, 5, r);
   q = new BigInteger(bitNums, 5, r);

   while (p.Equals(q))
      q = new BigInteger(bitNums, 5, r);

   //n merupakan hasil kali p dan q yang
      menjadi bilangan modulus
   n = p.multiply(q);

   //m merupakan nilai dari fungsi
      euler n
   BigInteger m = p.subtract(one).
      multiply(q.subtract(one));

   //e merupakan kunci publik yang
      relatif prima terhadap m
   e = new BigInteger(bitNums / 2, r);
   if (e.gcd(two).intValue() != 1)
      e = e.add(one);

   while (m.gcd(e).intValue() > 1)
      e = e.add(two);

   //d merupakan kunci privat yang
      dikalikan e kongruen dengan
      1 (mod m)
   d = e.modInverse(m);
}
```

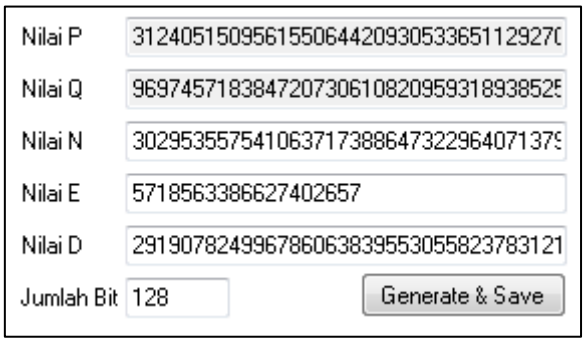
Fungsi ini pertama-tama akan menggenerate dua buah bilangan prima *p* dan *q* yang bertype BigInteger. Hasil generate ini kemudian akan dipergunakan untuk menggenerate nilai *n*, *e* dan *d*. Nilai n diperoleh dari perkalian *p* dan *q*. Sedangkan nilai *e* dan *d* merupakan pasangan nilai yang berkorespondensi dengan φ, dimana gcd(*e*, φ) = 1 dan *d* adalah invers modulo φ dari *e*.

Fungsi enkripsi-dekripsi algoritma RSA:

```
public string Encrypt(string input,
   BigInteger n, BigInteger key)
{
   int[] tmp = toArrInt(input);
   string retval = "";
   BigInteger m;

   //melakukan enkripsi blok-blok kecil
   for (int i = 0; i < tmp.Length; i++)
   {
      m = new BigInteger(tmp[i] + "");
      m = m.modPow(key, n);
      retval += (m.toString(16) + " ");
   }

   retval = retval.Substring(0,
      retval.Length - 1);

   return retval;
}
```

Fungsi enkripsi-dekripsi merupakan fungsi yang sama. Yang membedakan proses enkripsi atau dekripsi yang dilakukan bergantung pada nilai kunci yang dimasuk-

kan. Bila ingin mengenkripsi message hash, dipergunakan kunci private d sedangakan bila ingin mendekripsi ulangnya dipergunakan kunci publik e.



Gambar diatas merupakan hasil screenshoot penggunaan algoritma RSA untuk memperoleh kunci publik dan kunci privat.

5. KESIMPULAN

Terdapat banyak algoritma kirptografi kunci publik yang dapat diimplementasikan pada tanda tangan digital. Diantara banyak algoritma tersebut, RSA merupakan salah salah satu algoritma yang paling sering digunakan. Namun, penggunaan RSA pada algoritma tanda tangan digital untuk mendapatkan kunci publik dan kunci privat dinilai sudah tidak terlalu aman.

RSA merupakan algoritma kriptografi yang memanfaatkan randomisasi yang sulit ditembus karena membutuhkan perhitungan yang melibatkan bilangan besar. Namun, saat ini telah banyak serangan yang ditujukan pada RSA, misalnya *common modulus attack* dan *cyling attack.*Oleh karena itu, untuk tetap dapat menjamin faktor keamanan pada tanda tangan digital, sebaiknya digunakan algoritma kunci publik lain yang belum dapat ditembus sistem keamanannya.

DAFTAR REFERENSI

[1] Menez A, Oorschot, Vanstone S. *Handbook of Applied Cryptography*. 1996. CRC Press.

[2] Goldwasser, Shafi et al. *A Digital Signature Scheme Against Adaptive Chosen Message Attack.* 1988. SIAM Journal on Computing.

[3] Munir, Rinaldi. 2005. *Diktat Kuliah IF3038 Kriptografi*, Program Studi Teknik Informatika ITB: Bandung.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 17 Mei 2010

Muhammad Luthfi
13507129