Threshold Signature dengan Menggabungkan Algoritma RSA dan *Threshold Scheme*

Hendra Hadhil Choiri (135 08 041)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
hendra_h2c@students.itb.ac.id

Abstrakt— **Tanda tangan digital (*digital signature*) efektif dalam menjaga keaslian suatu dokumen milik seseorang. Namun, mungkin terjadi satu dokumen merupakan hak milik dari beberapa pihak sekaligus. Salah satu solusi yaitu tiap pihak membubuhkan tanda tangan digitalnya. Namun hal ini tidak dapat dilakukan jika ada pihak yang tidak bisa hadir untuk memberikan tanda tangan. Selain itu, akan sangat tidak efisien jika terdapat banyak tanda tangan sekaligus dalam satu dokumen.**

Untuk kasus seperti itu, solusinya adalah menggunakan tanda tangan ambang (*threshold signature*). Jadi, saat penandatanganan, tiap pihak mendapatkan kunci parsial yang dibangkitkan berdasarkan kunci privat utama *digital signature* tersebut. Tiap pihak tidak bisa mengetahui kunci dari pihak lain. Untuk penandatanganan, cukup dibutuhkan jumlah pihak minimum (misal t, t ≤ banyak pihak). Dengan menggabungkan t kunci mereka melalui suatu formulasi, dapat diperoleh kunci utama untuk membangkitkan tanda tangan digital. Penandatanganan tidak bisa dilakukan jika banyak pihak yang yang hadir kurang dari t. Lalu untuk verifikasi, cukup menggunakan kunci publik dari kunci privat utama.

Dalam makalah ini, *threshold signature* diimplementasikan dengan menggabungkan algoritma RSA dan *threshold scheme*.

***Kata Kunci*— tanda tangan digital, *threshold signature*, RSA, *threshold scheme*, dokumen**

I. PENDAHULUAN

Saat ini, berbagai teknologi digital sudah berkembang. Banyak hal di dunia nyata yang sudah didigitalkan ke dalam bentuk file, mulai dari dokumen, gambar, hingga perangkat lunak. Permasalahan yang muncul adalah mengenai kepemilikan dari suatu data digital. Memang, file digital bisa dengan mudah disalin oleh orang lain sehingga dapat dipermasalahkan siapa pemilik sebenarnya. Dalam ilmu kriptografi telah dibuat solusinya, yakni menggunakan tanda tangan digital untuk otentikasi kepemilikan suatu file/dokumen.

Namun, tanda tangan digital pada umumnya hanya efektif untuk menjaga keaslian satu dokumen milik satu orang saja. Sedangkan pada kenyataannya bisa jadi satu file/dokumen merupakan milik bersama dari beberapa pihak sekaligus. Solusi paling mudah yaitu setiap pihak melampirkan tanda tangan digitalnya masing-masing di dokumen. Dengan metode seperti itu, berarti semua pihak harus hadir saat penandatanganan.

Untuk menyelesaikan masalah tersebut, tanda tangan digital dapat dikombinasikan dengan *threshold scheme* (secret sharing *scheme*) menjadi *threshold signature*.[1] Dengan kombinasi tersebut, pada saat penandatanganan cukup dibutuhkan jumlah pihak minimum (misal t, di mana t ≤ banyak pihak). Penandatanganan tidak bisa dilakukan jika banyak pihak yang yang hadir kurang dari t. Sedangkan untuk verifikasi, cukup dibutuhkan sebuah kunci publik tunggal.

II. ALGORITMA KUNCI PUBLIK RSA

RSA adalah salah satu jenis kriptografi kunci publik yang dikembangkan oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1978.[3] Kunci publik berarti kuncinya tidak simetris, terdiri dari kunci publik untuk enkripsi dan kunci privat untuk dekripsi.

A. Pembangkitan Kunci

Perumusan algoritma RSA didasarkan pada teorema Euler yang menyatakan bahwa jika a relatif prima dengan n, maka berlaku:

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

dengan ϕ(n) adalah fungsi Euler, yakni banyaknya bilangan dari 1,2,3, …,n yang relatif prima terhadap n.

Selanjutnya, berikut ini adalah prosedur pembangkitan kunci privat dan kunci publik:

1. Pilih p dan q, yakni sembarang bilangan prima
2. Hitung n = p . q
3. Hitung m = (p-1)(q-1). Perhatikan bahwa m = ϕ(n), karena bilangan dari 1 hingga pq terdapat sebanyak q bilangan yang habis dibagi p, sebanyak p bilangan yang habis q, dan 1 bilangan yang habis dibagi pq,

jadi sisanya relatif prima terhadap pq, yakni sebanyak pq – p – q +1 = (p-1)(q-1).

4. Pilih kunci publik e, yang relatif prima terhadap m.
5. Bangkitkan kunci privat d, yaitu invers dari e (yakni yang memenuhi e . d ≡ 1 (mod ϕ(n)).
6. Akhirnya sudah didapatkan <e,n> sebagai kunci publik dan <d,n> sebagai kunci privat.

B. Enkripsi dan Dekripsi

Setelah memiliki kunci publik dan kunci privat, enkripsi dan dekripsi sudah dapat dilakukan terhadap plainteks. Namun, plainteks harus dikonversi dulu menjadi kumpulan blok bilangan bulat yang kurang dari n. Misal bloknya p_1, p_2, … Lalu, fungsi enkripsi untuk p_i yaitu:

$$c_i = p_i^e \bmod n$$

Sedangkan untuk dekripsi, gunakan fungsi:

$$p_i = c_i^d \bmod n$$

Fungsi tersebut dapat mengembalikan c*i* menjadi p*i* dengan memanfaatkan persamaan ed ≡ 1 (mod ϕ(n)), yang berarti ed = k ϕ(n) + 1. Serta dengan teorema Euler, diperoleh bahwa $p_i^{\phi(n)} \equiv 1 \pmod n$. Akhirnya didapat hasil berikut:

$$c_i^d \equiv (p_i^e)^d \equiv p_i^{ed} \equiv p_i^{k\,\phi(n)+1} \pmod n$$
$$\equiv (p_i^{\phi(n)})^k . p_i \equiv 1 . p_i \equiv \mathbf{p_i} \pmod n$$

III. SECRET SHARING *SCHEME*

A. Definisi

Secret Sharing adalah metode untuk mendistribusikan sebuah rahasia kepada sekelompok partisipan, sehingga masing-masing memperoleh sebagian rahasia. Rahasia hanya dapat direkonstruksi kembali saat sejumlah partisipan minimum menyatukan bagian rahasianya masing-masing.[2] Pada skema ini, terdapat beberapa terminologi penting yaitu:[3]

- ✓ Secret: data/informasi rahasia (password, kunci, PIN, pesan, file, dsb).
- ✓ Secret direpresentasikan sebagai sebuah integer M.
- ✓ Share: hasil pembagian secret
- ✓ Dealer: pihak yang melakukan pembagian secret
- ✓ Partisipan: orang yang memperoleh share.

B. Shamir's Threshold Scheme

Salah satu secret sharing *scheme* yang paling terkenal yaitu skema ambang (*threshold scheme*) yang ditemukan oleh Shamir. Misalkan t, w adalah bilangan bulat positif dengan t ≤ w. *Threshold scheme* (t, w) adalah metode pembagian pesan M kepada w partisipan sedemikian sehingga sembarang himpunan bagian yang terdiri dari t partisipan dapat merekonstruksi M, tetapi jika kurang dari t maka M tidak dapat direkonstruksi.[3]

Berdasarkan slide kuliah kriptografi [3], algoritmanya adalah sebagai berikut:

1. Pilih suatu bilangan prima *p* yang harus lebih besar dari semua kemungkinan nilai pesan *M* dan juga lebih besar dari jumlah *w* partisipan. p ini akan dijadikan modulus untuk semua perhitungan. p harus bilangan prima untuk memastikan tiap bilangan memiliki invers.

2. Selanjutnya, pilih $t - 1$ buah bilangan bulat dalam modulus *p* secara acak, misalkan $s_1, s_2, \ldots, s_{t-1}$, dan bentuk suatu polinomial:

$$s(x) \equiv M + s_1 x + s_2 x^2 + \ldots + s_{t-1} x^{t-1} \pmod p$$

sedemikian sehingga $s(0) \equiv M \pmod p$. s(x) ini disebut penulis sebagai **fungsi ambang.**

3. Untuk *w* partisipan, tentukan w bilangan bulat berbeda dalam modulus p, misal
$x_1, x_2, \ldots, x_w \pmod p$
dan setiap orang memperoleh *share* (x_i, y_i) di mana $y_i \equiv s_i(x_i) \pmod p$. Misalnya, untuk *w* orang kita memilih $x_1 = 1, x_2 = 2, \ldots, x_w = w$.

4. Misalkan *t* orang partisipan akan merekonstruksi *M*, dengan *share* masing-masing (x_1, y_1), (x_2, y_2) …, (x_t, y_t). Substitusikan setiap (x_k, y_k) ke dalam polynomial s(x), yang berarti:

$$y_k \equiv M + s_1 x_k^1 + \ldots + s_{t-1} x_k^{t-1} \pmod p, 1 \le k \le t$$

5. Misalkan $s_0 = M$, maka dapat ditulis ulang sistem persamaan ke dalam bentuk matriks:

$$\begin{pmatrix} 1 & x_1 & \cdots & x_1^{t-1} \\ 2 & x_2 & \ldots & x_2^{t-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_t & \ldots & x_t^{t-1} \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ \vdots \\ s_{t-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_t \end{pmatrix}$$

Selesaikan sistem persamaan linier di atas untuk memperoleh $s_0 = M$.

Dalam implementasi skema di atas, *p* tidak perlu rahasia, tetapi polinom *s*(*x*) harus dirahasiakan.

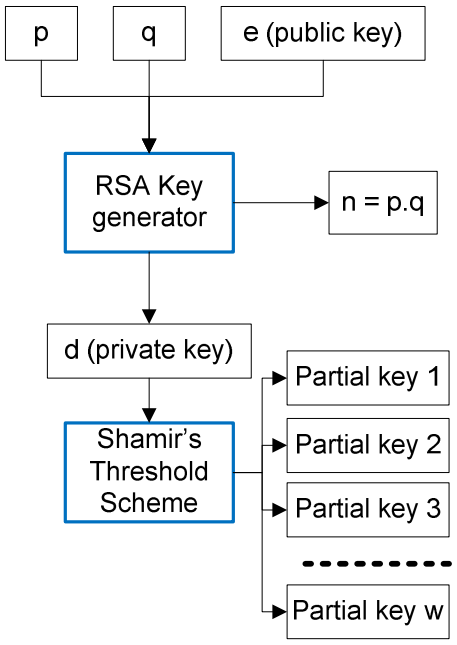
IV. IMPLEMENTASI

A. Protokol Threshold Signature

Salah satu solusi dari tanda tangan digital pada dokumen dengan banyak pemilik yaitu dengan *threshold signature* yang merupakan gabungan dari digital *signature* dan *secret sharing scheme*. Protokol dari *threshold signature* ini hampir mirip dengan protokol tanda tangan digital biasa, hanya saja kunci privat yang terbentuk dishare ke pihak-pihak pemiliknya menggunakan skema ambang.

Penulis telah mengimplementasikan *threshold scheme* ini dengan algoritma RSA sebagai enkripsi-dekripsi kunci publik, SHA-256 untuk membuat message digest, dan *threshold scheme* untuk melakukan share kunci. Protokolnya dapat dilihat melalui diagram-diagram berikut:

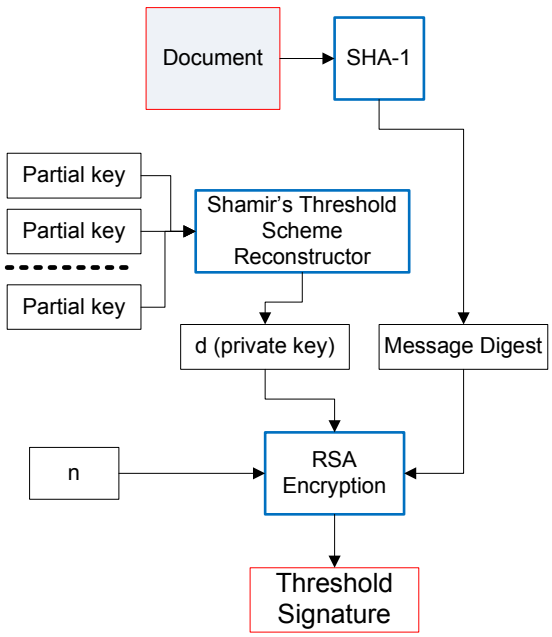
Protokol pembangkitan kunci



Pada pembangkitan kunci, seorang dealer memberikan input p, q, dan e yang digunakan untuk membangkitkan kunci publik dan kunci privat RSA. Akhirnya dihasilkan n, di mana pasangan <e,n> adalah kunci publik dari *threshold signature*, dan boleh disebarkan untuk umum. Kunci public ini akan digunakan untuk verifikasi.

Sedangkan d yang merupakan komponen kunci privat, dishare kepada w partisipan menggunakan Shamir's *threshold scheme*. Pada diagram di atas, terbentuk w buah kunci parsial yang akan dibagikan kepada w partisipan. Dalam membuat kunci parsial ini, ditentukan pula bilangan prima p yang dijadikan modulo perhitungan, serta koefisien untuk pembuatan fungsi ambang. Saat pembagian kunci parsial, harus dipastikan masing-masing pihak hanya mengetahui kunci parsial miliknya sendiri.

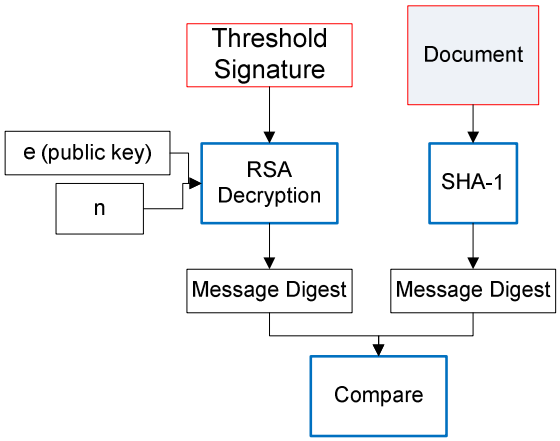
Protokol pembangkitan *threshold signature*



Pada diagram di atas, terlihat bahwa kunci privat untuk mengenkripsi message digest dari dokumen diperoleh dengan cara merekonstruksi kunci-kunci parsial dari partisipan minimum (misal t partisipan). Setelah kunci privat asli terbentuk, enkripsi RSA dapat dilakukan, dan tanda tangan digital berhasil dibuat.

Dengan protokol seperti ini, jika partisipan yang bisa hadir untuk menandatangani kurang dari t, *threshold signature* tidak dapat dibangkitkan karena belum cukup untuk merekonstruksi kunci privat.

Protokol verifikasi *threshold signature*



Terlihat bahwa protokol untuk verifikasi tepat sama dengan tanda tangan digital biasa, sehingga di ruang umum, tanda tangan digital pada dokumen ini tidak perlu dipermasalahkan apakah pemiliknya satu atau banyak orang, karena proses verifikasinya sama saja.

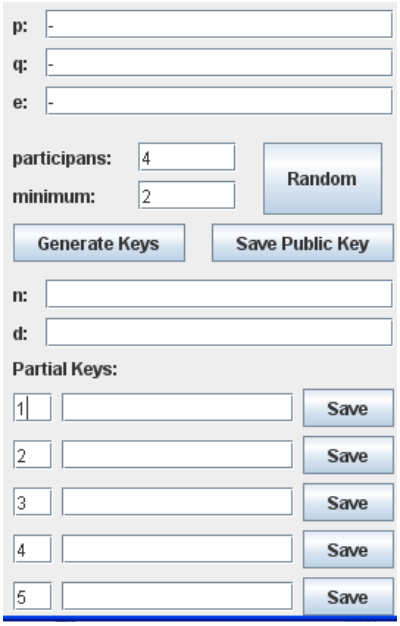
B. Program

Untuk menguji dan menganalisis *threshold signature*, penulis telah mengimplementasikannya dalam bentuk program dengan bahasa Java. Struktur data yang dibuat penulis terdiri dari kelas ThreSign sebagai kelas utama, kelas Key yang merupakan kunci-kunci RSA, serta PartialKey yang merupakan hasil share dari d. Berikut atribut-atributnya:

```
class ThreSign{
  int n_part;
  int minimum;
  Key K;
  BigInteger pp
  BigInteger[minimum] Koefisien;
  PartialKey[n_part] PK;
  byte[] dokumen;
}

class Key{
  BigInteger n;
  BigInteger e;
  BigInteger d;
}

class PartialKey{
  BigInteger x;
  BigInteger fx;
  BigInteger n;
}
```

Pembangkit Kunci

Berikut adalah antarmuka untuk membangkitkan kunci:



Terlihat bahwa untuk membangkitkan kunci, diperlukan p, q, e, banyak partisipan, dan share minimum. Untuk pengujian, p,q, dan e dipilih secara acak, di mana p dan q berukuran 40 bit dan e berukuran 30 bit. Bilangan prima pp dan koefisien untuk fungsi ambang juga dibangkitkan secara acak.

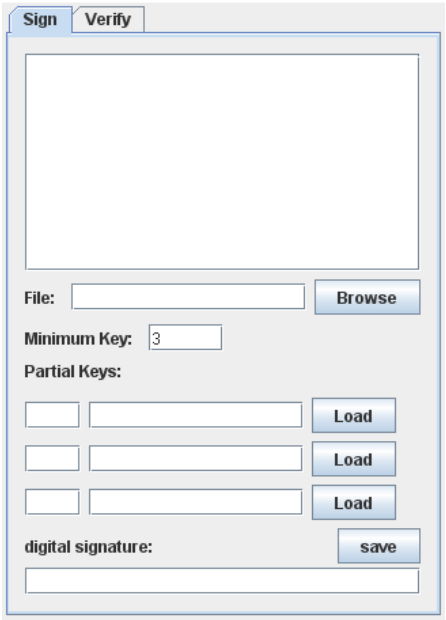
Berikut pseudocode untuk membangkitkan n, d, serta kunci-kunci parsial:

```
K.n ← p*q
BigInteger m ← (p-1)*(q-1)
K.d =  modInverse(K.e,K.m)

BigInteger Xn, FX
i traversal [0 .. n_part-1]
   Xn = PK.x[i];
   FX = d;
   j traversal [1 .. minimum]
      FX ← FX + Xn* Koefisien[j]
      Xn ← Xn*PK.x[i]
   endtraversal
   PK.fx[i] ← FX mod pp
   PK.n[i] ← K.n
endtraversal
```

Pada praktiknya, p, q, dan d tidak perlu ditampilkan di antarmuka. Kunci-kunci parsial juga seharusnya dirahasiakan oleh masing-masing partisipan. Namun, penulis menampilkannya untuk keperluan analisis.

Pembangkitan Tanda Tangan digital

Setelah kunci dibangkitkan, pemberian tanda tangan digital pada suatu dokumen/file sudah bisa dilakukan. Berikut antarmuka yang dibuat oleh penulis:



Dan untuk menghasilkan tanda tangan digital dari suatu dokumen, dibutuhkan kunci parsial sebanyak partisipan minimum. Untuk merekonstruksi kunci-kunci parsial menjadi d kembali, harus dicari koefisien-

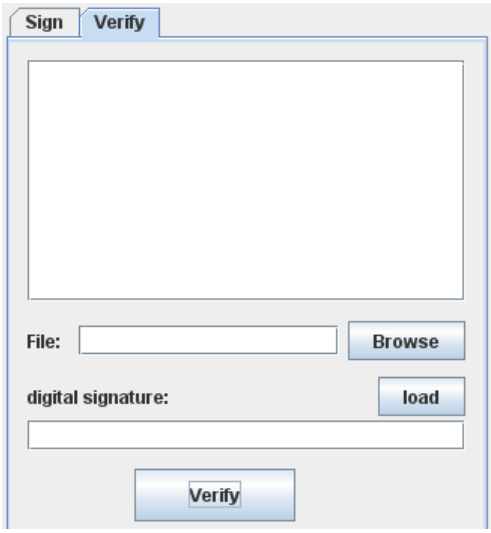
koefisien apda fungsi ambang. Penulis menggunakan *basis polinomial Lagrange*.[4] Berikut pseudocode untuk membangkitkan tanda tangan digital dengan RSA dan SHA-1:

```
PartialKey[minimum] Priv
BigInteger dd ← reconstruct(Priv)

byte[][] P ← partisi(SHA-1(dokumen), n)
byte[][] C

i traversal [0 .. banyak blok P]
   C[i] ← P[i]^dd mod n
Hex[] DS ← append(C)
```

Fungsi hash SHA-1 tidak diimplementasikan sendiri oleh penulis, tetapi memanfaatkan fungsi yang sudah disediakan oleh Java.

Verifikasi

Untuk verifikasi, dibandingkan antara message digest dokumen dengan dekripsi dari tanda tangan digital. Jika sama, maka dokumen itu sah. Jika tidak, maka dokumen sudah dipalsukan. Berikut antarmuka untuk verifikasi:



Dan pseudocode nya sama seperti verifikasi tanda tangan digital pada umumnya:

```
MD1 ← SHA-1(dokumen)

byte[][] C ← partisi(DS, n)
byte[][] P

i traversal [0 .. banyak blok C]
   P[i] ← C[i]^K.e mod K.n
MD2 ← append(P)

boolean isValid ← (MD1 = MD2)
```

C. Batasan

Dalam program yang dibuat penulis, terdapat beberapa hal yang harus dipenuhi agar dapat berjalan dengan baik. Batasan-batasan tersebut antara lain:

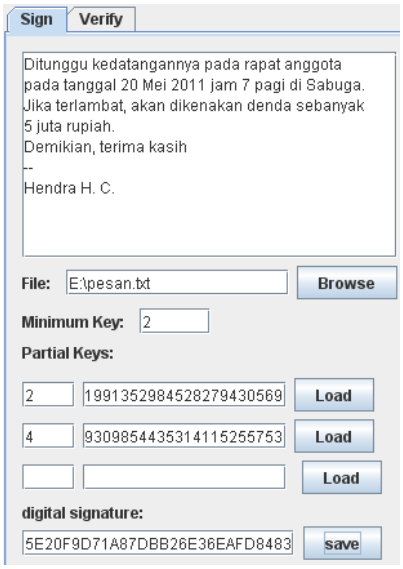
- ✓ Banyak nilai x yang diberikan pengguna harus sesuai dengan banyak partisipan, serta tidak boleh ada yang sama dalam modulo p
- ✓ Koefisien-koefisien untuk fungsi ambang dibangkitkan secara acak oleh program. Hal ini dimaksudkan agar kerahasiaan fungsi ambang lebih terjaga.
- ✓ Jika memilih p, q, dan e secara acak. Nilai yang dimunculkan diatur oleh program, yakni p dan q berukuran 40 bit, serta e berukuran 30 bit.
- ✓ Andaikan p, q, dan e dimasukkan secara manual, harus dipastikan bahwa yang dimasukkan adalah bilangan bulat. Program tidak menangani kesalahan input.
- ✓ Saat pengisian, banyak partisipan harus lebih banyak dari banyak kunci parsial minimum untuk tanda tangan.

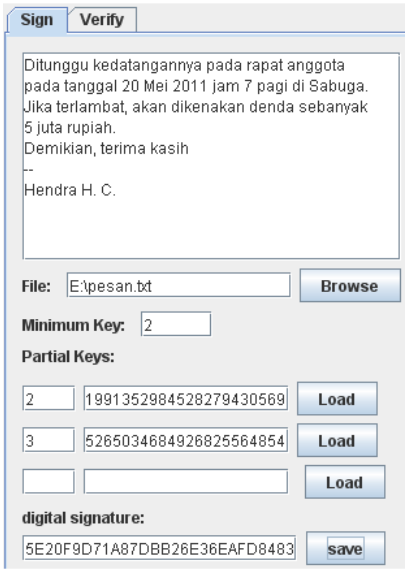
V. PENGUJIAN

Untuk pengujian, dipilih p,q, dan e secara acak. Ditentukan juga dokumen dimiliki oleh 4 orang, dan untuk penandatanganan cukup dibutuhkan kunci dari 2 partisipan. Berikut adalah kunci-kunci yang dihasilkan:

Dengan kunci-kunci parsial yang diperoleh tersebut, dapat dibuat tanda tangan. Berikut contoh tanda tangan digital yang diperoleh dengan menggunakan 2 dari 4

kunci parsial:



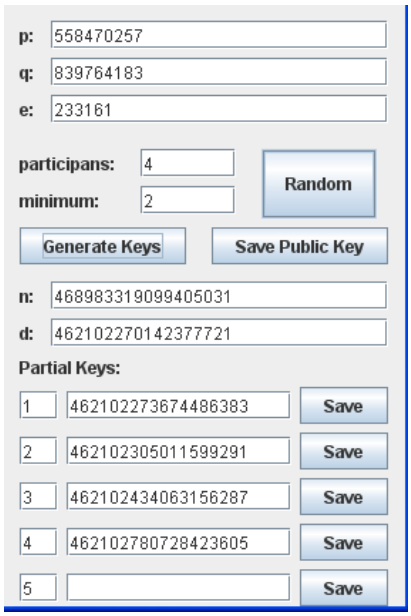


Terlihat bahwa apapun kombinasinya, tiap 2 kunci parsial selalu menghasilkan tanda tangan digital yang sama. Sedangkan untuk verifikasi, belum dilakukan pengujian oleh penulis. Makalah ini hanya menitikberatkan pada proses pembentukan *threshold scheme*.

VI. ANALISIS

A. Pemilihan koefisien fungsi ambang

Pada penerapan tanda tangan ambang (*threshold signature*), yang perlu diperhatikan adalah pemilihan koefisien-koefisien untuk fungsi ambang. Karena secret yang akan dishare adalah berupa kunci privat yang biasanya sangat besar (sekitar 200 bit). Jadi, jika koefisien yang dipilih terlalu kecil, akan terdapat celah keamanan yang dapat dimanfaatkan oleh kriptanalis untuk menemukan kunci privatnya. Misalnya pada contoh berikut, koefisien dipilih dengan ukuran 16 bit saja (integer). Yang terjadi justru kunci-kunci parsial yang dibangkitkan hampir sama angka-angka depannya. Terlihat juga bahwa angka-angka tersebut juga angka-angka pada kunci privat d.



Sedangkan jika koefisien terlalu besar (melebihi pp) akan kurang bagus juga. Karena akan dimoduluskan oleh pp sehingga ada kemungkinan tetap menjadi kecil. Untuk itu, di program yang penulis buat, koefisien dibangkitkan secara acak dengan mempertimbangkan ukuran dari kunci privat.

B. Kelebihan Threshold Scheme

Berikut ini adalah beberapa kelebihan dari konsep *threshold signature*:

- ✓ Satu dokumen bisa menjadi hak milik jadi beberapa pihak hanya dengan sebuah tanda tangan digital.
- ✓ Satu pihak tidak dapat mengambil alih kepemilikan

dokumen secara curang, karena membutuhkan pihak lain untuk melakukan tanda tangan.

- ✓ Dalam menandatangani dokumen, tidak harus dilakukan oleh semua pihak. Namun, hanya beberapa saja, tergantung banyak partisipan minimum untuk rekonstruksi kunci privat.
- ✓ Dapat mencegah pemaksaan tanda tangan, karena tanda tangan tidak bisa dibangkitkan hanya oleh satu pihak.
- ✓ Cara verifikasi sama saja dengan verifikasi tanda tangan digital yang lain.
- ✓ Konsepnya sederhana dan waktu pemrosesan tidak jauh berbeda dengan pembuatan tanda tangan digital biasa. Hanya saja kunci privatnya dishare.

C. Kelemahan Threshold Scheme

Di balik beberapa kelebihan di atas, dalam *threshold scheme* juga ditemukan beberapa kelemahan sebagai berikut:

- ✓ Saat pembagian kunci, dibutuhkan dealer yang dapat dipercaya sehingga kunci privatnya tidak bocor.
- ✓ Saat penandatanganan, sulit membuat prosedur agar suatu pihak tidak mengetahui kunci parsial milik pihak lain.
- ✓ Pemilihan kunci dan koefisien yang kurang tepat dapat memberikan celah keamanan yang dapat dimanfaatkan kriptanalis.

D. Aplikasi Threshold Scheme

Threshold scheme dapat diterapkan untuk berbagai hal, biasanya untuk keamanan dan hak milik dokumen. Berikut beberapa contoh penggunaannya:

- ✓ Membangkitkan tanda tangan pada dokumen dengan lebih dari satu dokumen. Sesuai dengan latar belakang pembuatan makalah ini.
- ✓ Untuk membuat pemegang kunci cadangan. Misal pemilik dokumen hanya satu orang, tetapi dia mempercayakan beberapa orang untuk memiliki kunci juga walaupun dalam bentuk share. Sehingga untuk mengetahui kunci privat sebenarnya, beberapa orang harus menggabungkan kunci parsial masing-masing.

VII. KESIMPULAN

Dari paparan di makalah ini, dapat disimpulkan bahwa *threshold signature* cukup efektif dalam pembuatan tanda tangan terhadap dokumen milik bersama. Implementasi *threshold signature* dari kombinasi algoritma RSA, SHA-256, dan Shamir's *threshold scheme* sudah cukup baik untuk menciptakan tanda tangan yang kokoh dan aman.

Agar lebih aman, pembagian kunci diserahkan kepada dealer yang dapat dipercaya, serta pemilihan p, q, e, dan koefisien harus dilakukan dengan bijak untuk keamanan yang optimal.

REFERENSI

Nadi Bozkurt, ilker, *Practical Threshold Signatures with Linear Secret Sharing Schemes*, Bilkent University.

Wikipedia, *Secret Sharing*, URL http://en.wikipedia.org/wiki/Secret_sharing, 18 April 2011

Munir, Rinaldi. 2011. *Skema Pembagian Data Rahasia* (Slide Kuliah). Bandung: Teknik Informatika ITB.

Wikipedia, Lagrange Polynomial, URL http://en.wikipedia.org/wiki/Lagrange_polynomial, 8 Mei 2011

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, 9 Mei 2011

Hendra Hadhil Choiri
135 08 041