Verifikasi Secret Sharing dengan Skema Feldman

Pembahasan dan Analisis Skema Feldman pada *Verifiable Secret Sharing*

Edwin Wijaya - 13513040
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
edwinwijaya1994@gmail.com

***Abstrak*— *Secret sharing* adalah salah satu metode pembagian pesan rahasia (*secret*) kepada sejumlah partisipan dengan syarat pesan rahasia tersebut dapat direkonstruksi jika jumlah partisipan yang hadir memenuhi skema ambang. Sebuah pesan rahasia akan didistribusikan oleh *dealer* kepada partisipan. Hasil pembagian pesan rahasia disebut share. Untuk menjamin *dealer* membagikan share yang benar (*valid*) maka diperlukan proses verifikasi *secret sharing*. Jika *dealer* membagikan *share* yang *invalid*, maka pesan rahasia tidak dapat direkonstruksi. Metode ini sering juga disebut sebagai *verifiable secret sharing* (VSS). Pada makalah ini verifikasi akan menggunakan skema Feldman yang merupakan perluasan skema Shamir pada pembagian pesan rahasia.**

Kata kunci—secret sharing, verifikasi, skema Shamir, skema Feldman, secret, share, dealer, partisipan, VSS

I. PENDAHULUAN

Pada era digital ini, kebutuhan akan data dan informasi menjadi semakin penting dan memegang peranan yang cukup besar dalam kehidupan manusia. Berbagai aspek kegiatan manusia mulai dari bisnis, medis, perbankan, transportasi, pemerintahan, militer, ekonomi, dan media sosial semuanya menggunakan data dan informasi sebagai salah satu komponen utama.

Kebutuhan dan penggunaan informasi yang masif ini tentunya perlu diikuti oleh aspek keamanan (*information security*). Ada 3 komponen yang harus dipenuhi terkait *information security* yaitu *confidentiality*, *integrity*, dan *availability* (biasa disingkat sebagai CIA). *Confidentiality* berarti menjamin akses terhadap data hanya diberikan kepada pihak yang memang memiliki hak akses. *Intergrity* berarti menjamin kebenaran data baik secara akurasi maupun kelengkapan data secara keseluruhan. Hal ini juga termasuk menjamin data yang tersimpan tidak dimodifikasi oleh pihak yang tidak bertanggung jawab. Sedangkan *availability* berarti menjamin data dapat selalu diakses kapanpun dibutuhkan. Pada umumnya, keamanan data terkait dengan proses penyimpanan (*data storage*) dan proses pengiriman (*data transmission*).

Beberapa jenis data yang penyimpanannya memerlukan keamanan yang cukup tinggi antara lain adalah penyimpanan kunci dekripsi, kode-kode peluncuran senjata / misil militer, akun perbankan, dan masih banyak lagi. Salah satu cara untuk menjamin keamanan penyimpanan data antara lain dengan melakukan enkripsi terhadap data yang disimpan.

Namun penggunaan enkripsi ternyata belum cukup untuk menjamin keamanan data yang disimpan terutama terkait *confidentiality* dan *availability* yang sulit untuk dipenuhi karena kunci enkripsi/dekripsi harus disimpan secara aman tetapi tetap mudah untuk diakses oleh pemilik pesan dan pihak yang mempunyai hak.

Untuk mengatasi permasalahan ini maka dibuat metode baru yang disebut pembagian pesan rahasia atau disebut sebagai *secret sharing*. Dengan metode ini, sebuah pesan rahasia akan dibagi menggunakan fungsi tertentu kemudian hasilnya (*share*) didistribusikan ke setiap pemilik pesan (partisipan) yang bersesuaian. Pesan rahasia dapat direkonstruksi jika jumlah partisipan yang ingin mengetahui pesan rahasia memenuhi jumlah minimum tertentu (skema ambang).

Pesan rahasia (misalnya berupa kunci enkripsi/dekripsi) akan dibagikan oleh *dealer* ke setiap partisipan sesuai dengan hasil keluaran fungsi yang memang ditujukan untuk melakukan pendistribusian pesan. Namun permasalahan yang mungkin muncul adalah bagaimana partisipan dapat yakin bahwa *share* yang diterimanya *valid* dan dapat digunakan untuk merekonstruksi pesan. Oleh karena itu, pada makalah ini akan dibahas sebuah metode/skema pembagian pesan rahasia yang memungkinkan setiap partisipan melakukan verifikasi terhadap *share* yang diterimanya apakah *valid* atau *invalid*. Skema pembagian pesan rahasia terverifikasi (*verified secret sharing* / VSS) yang akan dibahas dan dianalisis adalah skema Feldman yang merupakan perluasan skema Shamir (skema pembagian pesan rahasia yang pertama kali ditemukan).

II. DASAR TEORI

A. Secret Sharing

Secret sharing atau pembagian pesan rahasia adalah sebuah metode yang digunakan untuk membagi sebuah pesan rahasia berdasarkan fungsi tertentu (misalnya fungsi polynomial pada skema Shamir) yang kemudian diberikan kepada masing-masing partisipan sesuai dengan keluaran fungsi tersebut untuk setiap partisipan. Untuk memahami *secret sharing* diperlukan pemahaman terhadap istilah yang sering muncul berikut ini :

1. *Secret* : pesan, data, atau informasi rahasia yang direpresentasikan sebagai sebuah integer
2. *Share* : hasil pembagian *secret* berdasarkan fungsi tertentu
3. *Dealer* : pihak yang bertanggung jawab untuk membagi *secret* menjadi sejumlah *share*
4. Partisipan : orang yang berhak untuk menerima *share*

Dengan menggunakan *secret sharing* sebuah pesan rahasia (*secret*) dapat dibagi ke dalam sejumlah *share* sesuai dengan jumlah partisipan. *Share* ini digunakan untuk merekonstruksi pesan rahasia (mendapatkan pesan rahasia secara utuh). Pesan rahasia dapat direkonstruksi jika dan hanya jika jumlah partisipan yang ingin merekonstruksi pesan rahasia (yang memiliki *share*) mencapai jumlah minimum tertentu, misalnya t. Skema ini disebut juga sebagai skema ambang yang ditemukan oleh Shamir. Secara umum skema ambang Shamir dapat digeneralisasi sebagai berikut :

> Misalkan *t*, *w* adalah bilangan bulat positif dengan t ≤ w. Skema ambang (*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

Penggunaan *secret sharing* biasanya untuk informasi/pesan yang sangat penting dan rahasia, misalnya kunci enkripsi/dekripsi, kode peluncuran senjata / roket, dan informasi yang berkaitan dengan akun bank. Informasi seperti ini harus disimpan dan dijaga kerahasiaanya. Metode yang sudah lama digunakan adalah metode enkripsi yang bertujuan untuk membuat pesan tidak dapat dinterpretasi secara langsung. Akan tetapi dengan metode enkripsi, pemilik pesan harus menyimpan kunci dekripsi di tempat yang aman. Sesuai dengan prinsip keamanan informasi, setidaknya ada 2 hal yang harus dipenuhi yaitu *confidentiality* dan *availability.*

Untuk menjamin *confidentiality*, pengguna menyimpan kunci hanya di satu media penyimpanan saja. Namun hal ini tentu tidak dapat menjamin *availability* karena kunci hanya dapat diakses dari satu media penyimpanan saja. Untuk mengatasi hal ini, biasanya pengguna menduplikasi kunci dan menyimpannya di sejumlah media penyimpanan sehingga *availability* dapat terjamin. Akan tetapi solusi ini dapat meningkatkan risiko bocornya kunci ke pihak yang tidak diinginkan sehingga *confidentiality* tidak dapat dijamin lagi.

Dengan adanya *secret sharing* kedua syarat keamanan informasi dapat dipenuhi. Pembagian pesan rahasia (*secret sharing*) dapat diaktegorikan ke dalam 2 jenis yaitu *insecure secret sharing* dan *secure secret sharing*. Pada *insecure secret sharing*, sebuah pesan misalnya "kriptografer" akan dibagikan ke 3 responden. Pesan ini akan dipenggal menjadi "krip", "togr", dan "afer". Dengan skema ini, seorang partisipan yang hanya memiliki 1 *share* misalnya "krip" perlu menebak 8 huruf sisanya jika mengetahui bahwa panjang pesan adalah 12 huruf sehingga terdapat $26^8 \sim 308\,juta$ kemungkinan. Dengan semakin banyaknya partisipan yang memiliki *share*, maka semakin sedikit jumlah kemungkinan yang harus dicoba sampai menemukan pesan rahasia. Misalnya jika ada 2 partisipan, maka hanya ada $26^4 \sim 450\,ribu$ kemungkinan. Hal ini yang menyebabkan skema pembagian pesan rahasia sejenis ini kurang aman.

Pada *secure secret sharing*, misalnya terdapat pesan X, kunci public P_i , dan kunci privat Q_i. Setiap partisipan J akan diberikan pasangan $\{P_1(P_2(...(P_N(X)))), Q_j\}$. Dengan skema ini, partisipan 1 dapat membuka pesan pada *layer* 1 (menghilangkan P_1), partisipan 1 dan 2 dapat membuka pesan pada *layer* 1 dan *layer* 2. Seorang / sejumlah partisipan yang tidak memiliki N buah kunci tidak bisa mendekripsi pesan X. Dengan skema ini, pembagian pesan rahasia menjadi aman karena secara komputasi tidak mungkin menebak kunci privat dalam waktu yang realistis.

Salah satu bagian / proses dari *secret sharing* yang juga penting adalah bagaimana seorang partisipan dapat meyakini bahwa *share* yang diterimanya dari *dealer* adalah *valid* dan dapat digunakan untuk merekonstruksi pesan rahasia yang asli. Skema pembagian pesan rahasia yang memungkinkan partisipan melakukan verifikasi terhadap *share* yang diterimanya disebut juga sebagai *verifiable secret sharing* (VSS). Metode VSS yang dibahas pada makalah ini adalah skema Feldman.

B. Skema Shamir

Salah satu skema pembagian pesan rahasia yang paling pertama ditemukan adalah skema Shamir. Skema ini ditemukan oleh Adi Shamir pada 1979. Ide dari skema Shamir adalah dengan memanfaatkan persoalan interpolasi: untuk membentuk persamaan linier $y = a_0 + a_1x$ diperlukan 2 buah titik (x_1, y_1), (x_2, y_2) sedangkan untuk membentuk persamaan kuadratik $y = a_0 + a_1x + a_2x^2$ diperlukan 3 buah titik. Secara umum, untuk membentuk persamaan polynomial berderajat n diperlukan minimal n+1 buah titik.

Tujuan dari skema Shamir adalah membagi *secret* S kepada *w* partisipan menjadi sejumlah *share* S_1 , S_2 , … S_w dalam hal ini *share* berupa titik (x,y) sehingga berdasarkan skema ambang Shamir dapat ditarik kesimpulan sebagai berikut :

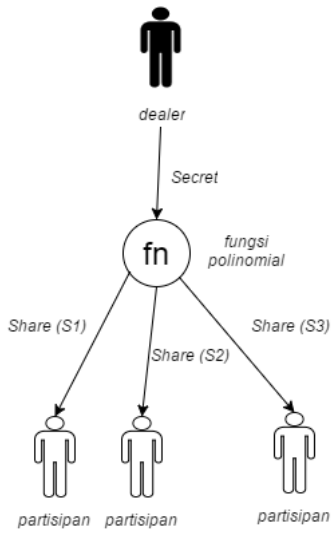
1. Sembarang himpunan bagian dari *share* yang banyaknya minimal *t* , yaitu jika terdapat minimal *t* buah titik maka persamaan polinomialnya dapat dibentuk dan secret S dapat direkonstruksi.

2. Sembarang himpunan bagian dari *share* yang jumlahnya *t-1* atau kurang, yaitu jika hanya terdapat *t-1* buah titik atau kurang, maka persamaan polinomial yang digunakan untuk membagi *secret* tidak dapat dibentuk sehingga *secret* tidak mungkin direkonstruksi.

Skema di atas sering ditulis sebagai skema (*t*, *w*).

Berikut adalah algoritma skema Shamir dalam pembagian pesan rahasia :

1. Pilih bilangan prima p, yang harus lebih besar dari semua kemungkinan nilai pesan M dan juga lebih besar dari jumlah w partisipan. Semua komputasi dihasilkan dalam modulus p.
2. Pilih secara acak $t - 1$ buah bilangan bulat dalam modulus p, misalkan $s_1, s_2, \ldots, s_{t-1}$, dan nyatakan polinomial:
   $s(x) \equiv M + s_1x + s_2x^2 + \ldots + s_{t-1}x^{t-1} \pmod p$
   sedemikian sehingga $s(0) \equiv M \pmod p$.
3. Untuk w partisipan, kita pilih *integer* berbeda, $x_1, x_2, \ldots, x_w \pmod p$ dan setiap orang memperoleh *share* (x_i, y_i) yang dalam hal ini :
   $y_i \equiv s_i(x_i) \pmod p$.
   Misalnya, untuk w orang kita memilih $x_1 = 1$, $x_2 = 2$, $\ldots$, $x_w = w$.



Gambar 1 – Skema Shamir dalam pembagian pesan rahasia

Berikut adalah contoh pembagian pesan rahasia dengan skema Shamir. Skema yang digunakan adalah skema(3, 8) :

1. Artinya: $w = 8$ partisipan, diperlukan minimal $t = 3$ partisipan untuk melakukan rekonstruksi pesan rahasia M.
2. Misalkan M = 190503180520 (*secret*)
3. Misalkan p = 1234567890133 (prima)
4. Pilih 3 – 1 = 2 buah bilangan acak, sebut s_1 dan s_2, untuk membentuk polinom:

   $s(x) \equiv M + s_1x + s_2x^2 \pmod p$
   maka persamaan polinomialnya adalah :
   $s(x) \equiv 190503180520 + 482943028839x + 1206749628665x^2 \pmod{1234567890133}$
   Perlu diketahui juga bahwa polinom harus dirahasiakan sehingga yang bersifat publik hanyalah nilai t, w, dan nilai modulus p.
5. Tiap partisipan memperoleh $(x, s(x))$. Misakan $x_1 = 1$, $x_2 = 2, \ldots, x_8 = 8$, maka, setiap orang memperoleh *share*:
   (1, 645627947891)
   (2, 1045116192326)
   (3, 154400023692)
   (4, 442615222255)
   (5, 675193897882)
   (6, 852136050573)
   (7, 973441680328)
   (8, 1039110787147)

Untuk merekonstruksi pesan diperlukan minimal sembarang t partisipan yang kemudian mensubstitusikan *share* masing-masing ke dalam persamaan polinomial. Dengan menggunakan metode interpolasi Langrange, persamaan polinomial dapat ditemukan dan *secret* yang merupakan konstanta dari persamaan polinomial dapat direkonstruksi.

Pemilihan koefisien untuk membentuk persamaan polinomial menjadi sangat penting, umumnya menggunakan bilangan yang sangat besar sehingga sulit untuk ditebak, bahkan membutuhkan waktu komputasi yang sangat lama sehingga keamanannya lebih terjamin. Pemilihan nilai modulus p juga perlu diperhatikan, yaitu dengan mimilih nilai p yang merupakan bilangan prima yang lebih besar dari nilai pesan M dan jumlah partisipan w.

C. Homomorphic Encryption

Homomorphic encryption adalah salah satu bentuk enkripsi yang memungkinkan sebuah hasil perhitungan / komputasi pada cipherteks jika dilakukan dekripsi maka hasil perhitungan tersebut nilainya akan sama jika perhitungan tersebut langsung dilakukan pada bentuk plainteksnya.

Penggunaan *homomorphic encryption* misalnya untuk membangun sebuah sistem yang terdiri dari sejumlah *service* yang saling berkaitan, namun pengguna tidak ingin memberikan data (plainteks) secara langsung kepada setiap *service* tersebut, sehingga proses komputasi yang dilakukan *service* dapat langsung dilakukan terhadap bentuk *cipherteks* nya tanpa mengubah hasil perhitungan ketika nantinya dilakukan dekripsi pada hasil perhitungan tersebut. Beberapa sistem yang membutuhkan *homomorphic encryption* antara lain sistem perpajakan dan sistem *voting*.

Homomorphic encryption ada yang bersifat *partially*, yaitu hanya beberapa jenis operasi saja yang bersifat *homomorfik*, misalnya hanya operasi penjumlahan, perkalian, dan fungsi kuadratik. Namun ada juga yang bersifat *fully homomorphic* yang memungkinkan semua jenis operasi perhitungan dapat bersifat *homomorphic*. Contoh pemanfaatan sifat *homomorphic* pada algoritma kunci publik RSA :

$$E(X_1).E(X_2) = X_1^e.X_2^e mod\ m = (X_1.X_2)^e\ mod\ m$$
$$= E(X_1.\ X_2)$$

dengan E(X) menyatakan enkripsi pada pesan X.

Contoh sifat *homomorphic* pada algoritma ElGamal :

$$E(M_1).E(M_2) = (g^{r1}, m_1.h^{r1}).(g^{r2}, m_2.h^{r2})$$
$$= (g^{r1+r2}, (m_1.m_2).h^{r1+r2})$$
$$= E(M_1.\ M_2)$$

dengan generator *g*, pesan *m*, kunci privat x, h = g^x, , dan r adalah bilangan acak dalam modulus q.

III. SKEMA FELDMAN

A. *Pembahasan*

Skema Feldman merupakan salah satu *verifiable secret sharing (VSS)* yang paling umum digunakan. Skema Feldman digunakan oleh partisipan untuk melakukan verifikasi terhadap *share* yang diterima dari *dealer* apakah *valid* atau *invalid*. Skema pembagian pesan rahasia ini merupakan perluasan skema Shamir yang dikombinasikan dengan sifat *homomorphic encryption*. Ide dasarnya adalah memanfaatkan sebuah fungsi *f* sehingga f(x+y) = f(x) · f(y). Kemudian setiap keluaran fungsi yaitu $f(a_0)$, $f(a_1)$,…$f(a_{k-1})$ akan diberikan ke partisipan yang bersesuaian, dengan $P(x) = a_0 + a_1x + \cdots + a_{k-1}x^{k-1}$ adalah fungsi polinomial yang digunakan pada skema Shamir. Parameter masukan pada fungsi *f* merupakan koefisien dari persamaan polinomial yang sering disebut sebagai *commitments* karena *dealer* menjamin kebenaran dari nilai koefisien-koefisien ini. Salah satu kandidat fungsi *f* yang memenuhi adalah $f(x) = a^x \ mod\ p$.

Untuk lebih jelasnya perhatikan algoritma / langkah – langkah dalam skema Feldman berikut :

1. *Dealer* membagi *secret* sesuai dengan metode Shamir's *secret sharing* dengan fungsi polynomial $P(x) = s_1x + s_2x^2 + \cdots + s_{t-1}x^{t-1}\ mod\ q$ yang menghasilkan *share* P(1), P(2), …
2. Definisikan sebuah group G dengan nilai modulus *p* (hasil fungsi polynomial dihitung dalam modulus *q*, sedangkan perhitungan *commitments* menggunakan modulus *p*) dan sebuah nilai g (generator) sehingga sulit untuk menghitung logaritma diskrit pada group G ini. Perlu diperhatikan juga bahwa pemilihan nilai *p* dan *q* harus memenuhi q | p-1 (*q* habis membagi *p-1*) serta *p* dan *q* keduanya adalah bilangan prima.
3. *Dealer* menghitung *commitments* (hasil keluaran fungsi g^x dengan *x* adalah koefisien dari fungsi polinomial *P(x)*) sehingga membentuk persamaan-persamaan berikut : $c_0 = g^s, c_1 = g^{s1}, c_2 = g^{s2}, \dots, c_t = g^{a_t}$
   Semua nilai *commitments* bersifat publik, namu nilai *g* hanya diketahui *dealer*.
4. Partisipan dapat melakukan verifikasi terhadap *share* yang diterima dengan cara mengecek apakah share P(i) dapat memenuhi persamaan berikut :

$$\alpha^{P(i)}\ mod\ p = \prod_{j=0}^{k-1} \alpha_j^{ij}\ mod\ p$$

dengan α = g^x adalah *commitments*.

B. *Pengujian*

Pada sub-bab ini akan diberikan contoh – contoh hasil pengujian skema Feldman menggunakan implementasi program sederhana berbasis Java (kode program dapat diakses di https://github.com/edwinwijaya94/FeldmanVSS). Program ini meminta input pengguna (partisipan) berupa ID dari partisipan dan share yang diterima oleh partisipan tersebut. Pada program ini, persamaan polinomial dan *share* yang *valid* seluruhnya ditampilkan untuk memudahkan pengecekan kebenaran program. Pada kondisi aslinya, partisipan hanya menerima *share* miliknya sendiri dan tidak mengetahui persamaan polinomial yang digunakan.

Berikut adalah contoh hasil pengujian :

Testcase 1 :

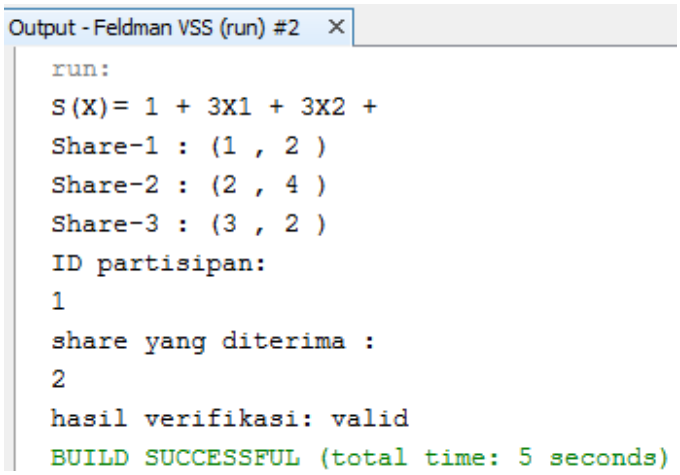
Skema Shamir : skema (3, 3)

Pesan rahasia (M) : 1

Persamaan polinomial : $P(x) = 1 + 3x + 3x^2\ mod\ 5$

Fungsi *commitments* : $\alpha = 3^x\ mod\ 11$

Hasil keluaran program :



Penjelasan :

Pada *testcase* ini, partisipan dengan ID 1 menerima *share* yang nilainya 2. Program menyatakan hasil verifikasi *valid* karena *share* yang diterima sesuai.

Testcase 2 :

Skema Shamir : skema (3, 3)

Pesan rahasia (M) : 1

Persamaan polinomial : $P(x) = 1 + 3x + 3x^2 \ mod \ 5$

Fungsi *commitments* : $\alpha = \ 3^x \ mod \ 11$

Hasil keluaran program :

```
Output - Feldman VSS (run) #2
run:
S(X)= 1 + 3X1 + 3X2 +
Share-1 : (1 , 2 )
Share-2 : (2 , 4 )
Share-3 : (3 , 2 )
ID partisipan:
2
share yang diterima :
3
hasil verifikasi: invalid
BUILD SUCCESSFUL (total time: 3 seconds
```

Penjelasan :

Pada *testcase* ini, partisipan dengan ID 2 menerima *share* yang nilainya 3. Program menyatakan hasil verifikasi *invalid* karena *share* yang diterima seharusnya bernilai 4 sehingga dalam hal ini, *dealer* memberikan *share* yang *invalid* kepada partisipan 2.

Testcase 3 :

Skema Shamir : skema (4, 4)

Pesan rahasia (M) : 2

Persamaan polinomial : $P(x) = 2 + 4x + 2x^2 + x^3 \ mod \ 5$

Fungsi *commitments* : $\alpha = \ 3^x \ mod \ 11$

Hasil keluaran program :

```
Output - Feldman VSS (run) #5
run:
S(X)= 2 + 5X1 + 2X2 + 1X3 +
Share-1 : (1 , 0 )
Share-2 : (2 , 3 )
Share-3 : (3 , 2 )
Share-4 : (4 , 3 )
ID partisipan:
3
share yang diterima :
1
hasil verifikasi: invalid
BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan :

Pada *testcase* ini, partisipan dengan ID 3 menerima *share* yang nilainya 1. Program menyatakan hasil verifikasi *invalid* karena *share* yang diterima seharusnya bernilai 2 sehingga dalam hal ini, *dealer* memberikan *share* yang *invalid* kepada partisipan 3.

IV. ANALISIS

A. *Analisis Performansi*

Performansi skema Feldman untuk VSS dan juga algoritma *homomorphic encryption* pada umumnya secara kompleksitas memori dan kompleksitas waktu masih dapat dikatakan realistis. Secara singkat, *dealer* pertama-tama akan mendistribusikan *share* ke masing-masing partisipan kemudian mendistribusikan *commitments*.

Berdasarkan kompleksitas memori, ukuran bit yang harus dikirimkan untuk mendistribusikan *share* dan commitments sebanding dengan *O(nk)* bit dengan *n* adalah jumlah partisipan dan *k* adalah ukuran *share* / *commitments*. Dalam hal ini dapat diasumsikan, *share* berukuran 1000 bit dapat dikatakan cukup aman karena sulit untuk mencari logaritma diskrit untuk bilangan sebesar itu.

Berdasarkan komplekitas waktu, bagian yang paling banyak membutuhkan waktu komputasi adalah menghitung *commitments* $\alpha = g^x \ mod \ p$. Perhitungan ini setidaknya sebanding dengan *x* kali operasi perkalian dan modulo untuk setiap bit k. $(g \ mod \ p \ . g \ mod \ p. \ldots \ g \ mod \ p)$. Kompleksitas waktunya bergantung pada kecepatan operasi perkalian yaitu *O(k log k)*. Karena *dealer* harus mendistribusikan ke *n* partisipan, maka kompleksitas waktu totalnya adalah

$$O(nk) \cdot O(k \ log \ k) = O(nk^2 \log k).$$

B. *Analisis Keamanan*

Aspek keamanan pada skema Feldman ditentukan oleh 2 faktor. Pertama, apakah skema yang dirancang dapat melakukan validasi secara tepat. Hal ini dijamin dengan memanfaatkan sifat *homomorphic encryption* pada persamaan *commitments* yang digunakan $\alpha = g^x \ mod \ m$ sehingga memenuhi sifat berikut $Encrypt(AB) = (AB)^x \ mod \ m = Encrypt(A) \cdot Encrypt(B)$.

Aspek keamanan yang kedua adalah memastikan bahwa setiap partisipan tidak dapat memperoleh informasi *s* dari setiap *commitments* $g^s \ mod \ m$ dengan *s* adalah konstanta dan koefisien dari fungsi polinomial. Jika partisipan dapat menemukan setiap nilai *s*, maka fungsi polinomial bocor dan *secret* tidak lagi bersifat privat. Untuk menjamin fungsi polinomial tetap rahasia, maka diperlukan nilai *g* yang cukup besar sehingga tidak mungkin menghitung nilai *s* jika diketahui nilai $g^s \ mod \ m$. Persoalan ini adalah persoalan menemukan logaritma diskrit yang juga menjadi kunci keamanan pada algoritma ElGamal. Persoalan serupa yang juga dapat

digunakan pada skema Feldman adalah ECDLP (*elliptic curve discrete logarithm problem*) yang menjadi kunci keamanan algoritma ECC.

V. SIMPULAN

Berdasarkan pembahasan, pengujian, dan analisis yang dilakukan, skema Feldman merupakan salah satu protokol yang dapat digunakan dalam *verifiable secret sharing* (VSS). Dengan menggunakan skema Feldman yang merupakan perluasan skema Shamir, partisipan dapat melakukan verifikasi terhadap *share* yang diterimanya apakah *valid* atau *invalid*. Skema Feldman juga memanfaatkan sifat *homomorphic encryption* sehingga perhitungan / komputasi tetap dapat dilakukan dalam bentuk terenkripsi untuk menjamin kerahasiaan. Secara kompleksitas waktu dan memori, skema Feldman juga cukup realistis sehingga dapat diterapkan dalam skema pembagian pesan rahasia.

REFERENSI

[1] Paul Feldman. A practical scheme for non-interactive verifiable secret sharing. In Proc. of the 28th IEEE Ann. Symp. on Foundations of Computer Science, pages 427–437. IEEE, October 1987.

[2] Theodore M. Wong, Chenxi Wang, Jeannette M. Wing. Verifiable Secret Redistribution for Threshold Sharing Schemes . CMU-CS-02-114, School of Computer Science, Carnegie Mellon University.

[3] http://profs.info.uaic.ro/~siftene/Feldman.pdf diakses pada 14 Mei 2016

[4] Michael Backes, Aniket Kate, Arpita Patra. Computational Verifiable Secret Sharing Revisited. (https://eprint.iacr.org/2011/281.pdf)

[5] Slide perkuliahan IF4020-Kriptografi
http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2012-2013/Skema%20Pembagian%20Data%20Rahasia%20(2013).ppt

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, 18 Mei 2016

Edwin Wijaya
13513040