Studi Dan Implementasi *Clustering* Penerima Kunci Dengan Metode *Shamir Secret Sharing Advanced*

Ir. Rinaldi Munir M.T[1], Addie Baratha[2]

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : rinaldi@informatika.org [1], if18067@students.if.itb.ac.id [2], addiehz@yahoo.com [2]

Abstrak

Suatu *secret* jika hanya dipegang oleh satu orang atau satu pihak akan meningkatkan resiko keamanan. Salah satu cara untuk mengatasi permasalahan ini adalah dengan memecah *secret* tersebut menjadi beberapa bagian. *Secret sharing scheme* merupakan suatu cara atau metode untuk membagi suatu rahasia, atau *secret*, menjadi beberapa bagian yang disebut *shares*, untuk dibagikan kepada sejumlah pihak yang disebut *participants*, dengan suatu aturan tertentu. Aturan yang dimaksud adalah struktur akses, yaitu aturan tentang siapa saja yang mendapatkan otorisasi untuk membentuk *secret* kembali. *Secret* yang dimaksud lebih khusus adalah kunci hasil suatu proses enkripsi. Makalah ini membahas tentang bagaimana melakukan pembagian *secret* dan pembangkitan kembali *secret* dengan menggunakan metode *Shamir Secret Sharing Advanced*. Metode ini juga mengatur bagaimana mengelompokkan (*clustering*) penerima kunci. Dalam suatu kelompok berlaku nilai ambang (*threshold value*), yaitu jumlah minimal *participants* yang dibutuhkan dalam suatu kelompok untuk membangkitkan *secret*.

***Kata kunci**: Secret, Shamir secret sharing scheme, participants, Shamir secret sharing scheme advanced,shares*, kunci, clustering, nilai ambang, struktur akses.

1. Pendahuluan

Untuk menjaga keamanan kunci hasil dari sistem kriptografi agar tidak hilang, disarankan untuk membuat sejumlah kunci cadangan. Namun resiko kerahasiaan kunci akan semakin besar dengan semakin banyaknya kunci cadangan yang dibuat. *Secret sharing* menangani masalah ini dengan membagi kunci menjadi beberapa bagian tanpa meningkatkan resiko kerahasiaan. *Secret sharing* juga menangani masalah pendistribusian kunci–kunci tersebut dengan hanya mengijinkan t dari n user dimana $t \leq n$ untuk melakukan pembentukan kunci awal.

Ide dari *secret sharing* adalah dengan membagi kunci rahasia menjadi beberapa bagian yang disebut *shares*, dan membagikannya kepada beberapa orang. Hanya *subset* dari orang – orang tersebut yang bisa atau diijinkan untuk membentuk kunci awal kembali.

Contoh kasus pengembangan metode Shamir adalah pembagian pemegang kunci menjadi 2 kelompok. Untuk membentuk kunci awal, misalnya diperlukan sedikitnya 2 shares dari kelompok A dan 3 shares dari kelompok B. Berapapun jumlah shares pada kelompok A, mereka tidak dapat membentuk kunci awal tanpa minimal 2 shares dari kelompok B, begitu juga sebaliknya.

2. *Threshold Scheme*

Skema ini bertujuan untuk membagi *secret S* menjadi n bagian $S_1, S_2, \ldots, S_n$ sedemikian sehingga:

1. Sejumlah k atau lebih *shares* dapat melakukan pembangkitan secret *S*.
2. Sejumlah k-1 atau kurang shares tidak dapat digunakan untuk membangkitkan *secret S* dan tidak menggambarkan informasi apapun tentang *secret S*.

Skema ini disebut *(k,n) threshold scheme*, dengan *k* disebut *threshold value* atau nilai ambang.

Threshold scheme yang efisien akan sangat membantu dalam proses manajemen kunci kriptografi. Untuk pengamanan data, dapat dilakukan dengan mengenkripsinya, tapi untuk melakukan pengamanan kunci kriptografi, diperlukan metode-metode yang berbeda lagi. Salah satu metode untuk pengamanan kunci kriptografi saat ini adalah dengan menyimpang kunci pada satu lokasi yang dijaga dengan baik, contohnya dalam komputer dengan tingkat keamanan yang tinggi. Namun metode ini mengandung resiko yang besar jika komputer mengalami *crash* atau *down*.

Dengan menggunakan skema *(k,n) threshold* dengan $n = 2k - 1$, didapatkan skema manajemen kunci yang aman. Proses pembangkitan kunci dapat dilakukan meskipun seandainya sejumlah $[n / 2] = k - 1$ dari n buah *shares* dihancurkan atau dicuri pihak yang tidak terotorisasi. Namun dengan

sejumlah [n / 2] = k – 1 buah *shares*, pihak yang tidak terotorisasi tersebut tidak dapat membangkitkan kunci atau *secret*.

Skema *threshold* cocok untuk aplikasi yang melibatkan sejumlah pihak yang tidak saling mempercayai atau memiliki kepentingan masing-masing. Dengan memilih parameter k dan n yang sesuai, dapat ditentukan kekuatan otoritas untuk melakukan pembentukan kunci dan kekuatan otoritas untuk menghalangi pembentukan kunci.

Sebagai contoh sederhana, sebuah secret S dibagi menjadi $n = m + 1$ buah shares sedemikian sehingga sejumlah n *shares* dapat membangkitkan kembali *secret*. Diasumsikan $S < p$ dengan p adalah bilangan prima. Dipilih $n - 1$ bilangan acak $S_1, S_2, \ldots, S_n$ yang berbeda-beda dari himpunan bilangan bulat $\mathbb{Z}_p$.

Kemudian dihitung S_n dengan persamaan P 2.1 berikut ini,

$$S_n = S - \sum_{i=1}^{n-1} S_i \pmod p \qquad \text{P 2.1}$$

yang berpadanan dengan persamaan P 2.2.

$$S = \sum_{i=1}^{n} S_i \pmod p \qquad \text{P 2.2}$$

Sehingga proses pembangkitan *secret S* dapat dilakukan jika sejumlah *n shares* terkumpul dengan penjumlahan sederhana. Namun jika hanya terkumpul $n - 1$ *shares*, atau hanya terdapat $S_1, S_2, \ldots, S_{n-1}$ *shares*, maka *shares-shares* tersebut tidak menggambarkan informasi apapun tentang *secret S*. Skema ini disebut *(n,n) threshold scheme.*

3. *Shamir Secret Sharing*

Shamir secret sharing scheme adalah *threshold scheme* berdasarkan interpolasi polinomial. Sejumlah k titik dalam ruang dua dimensi (x_1,y_1), (x_2,y_2), …, (x_k,y_k) dengan x_i yang berbeda-beda, akan membentuk tepat hanya satu persamaan polinomial $q(x)$ dengan derajat k – 1 sedemikian sehingga berlaku $q(x_i) = y_i$ untuk semua $1 \le i \le k$. Dari pernyataan tersebut dan tanpa mengurangi makna secara umum, dapat diasumsikan sebuah data D sebagai angka, akan dibagi menjadi beberapa bagian sejumlah n, maka dipilih sebuah persamaan polinomial berderajat k – 1 secara random seperti pada persamaan P 3.1,

$$q(x) = a_0 + a_1 x + \ldots + a_{k-1} x^{k-1} \qquad \text{P 3.1}$$

dengan $a_0 = D$, dan dilakukan penghitungan untuk tiap *shares* yang terbentuk sebagai persamaan P 3.2 berikut ini.

$$D_1 = q(1), \ldots, D_i = q(i), \ldots, D_n = q(n) \qquad \text{P 3.2}$$

Sebarang *subset* k dari D_i, dapat dicari koefisien persamaan polinomial $q(x)$ dengan melakukan interpolasi, dan kemudian menghitung $D = q(0)$. Namun, sejumlah k – 1 dari D_i, tidak dapat atau tidak cukup untuk melakukan perhitungan mendapatkan D.

Metode *shamir secret sharing scheme* menggunakan aritmatika modulo (*modular arithmatic*) sebagai pengganti *real arithmatic*. Sebagai gambaran, untuk suatu data D, dipilih sebuah bilangan prima p yang lebih besar D dan n. Koefisien $a_1, a_2, a_3, \ldots, a_{k-1}$ dalam persamaan $q(x)$ dipilih secara random dari himpunan bilangan bulat dalam $[0, p)$ dan nilai-nilai $D_1, D_2, \ldots, D_n$ dihitung menggunakan modulo p.

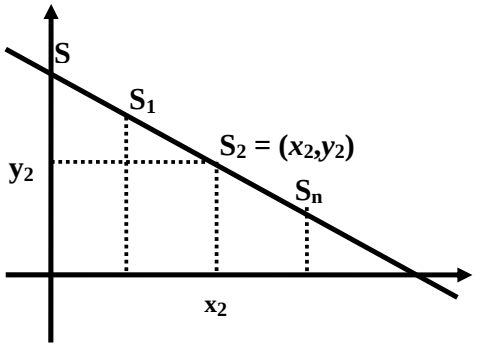
Misalnya sejumlah $k - 1$ dari n *shares* jatuh ke pihak yang tidak terotorisasi. Untuk setiap kandidat nilai D' dalam $[0, p)$, dapat dibangkitkan persamaan polinomial $q'(x)$ dengan derajat $k - 1$ sedemikian sehingga $q'(0) = D'$, dan $q'(i) = D_i$ untuk setiap k-1 parameter. Maka, sistem persamaan yang diperoleh akan mempunyai derajat yang berbeda dengan sistem persamaan asal (sistem persamaan untuk membangkitkan D), sehingga hasil yang diperoleh D' akan berbeda dengan D dan tidak ada informasi apapun yang diketahui tentang D.

Shamir secret sharing scheme secara matematis dibagi menjadi dua protokol, yaitu

1. *Sharing Protocol*
   Tujuan protokol ini adalah untuk membagi *secret S* kepada sejumlah n $P_1, P_2, \ldots, P_n$ *participant* sedemikian sehingga dibutuhkan sejumlah *t shares* untuk membangkitkan kembali *secret S*.
   a. *Dealer* membangkitkan persamaan polinomial secara random $f(x)$ dengan derajat $t - 1$ dengan persamaan P 3.3.

      $$f(x) = a_0 + a_1 x + \ldots + a_{t-1} x^{t-1} \qquad \text{P 3.3}$$

      Polinomial ini merupakan persamaan tidak tak-terhingga, dengan a_0 adalah *secret S* dan koefisien-koefisien lainnya dipilih secara random.
   b. *Dealer* memilih n titik random yang berbeda-beda, dengan $x_j \neq 0$, dan membagikannya secara rahasia kepada setiap *participants.* Untuk *participant* P_j akan mendapatkan $share_j(s) = (x_j, f(x_j))$ untuk $j = 1, 2, \ldots, n$.
2. *Reconstruction Protocol*
   Secara umum, proses *reconstruction* atau pembangkitan *secret S* dilakukan jika terdapat sejumlah *t shares* yang merupakan himpunan bagian dari *n shares*.
   a. Interpolasi persamaan polinom untuk mendapatkan persamaan polinomial yang unik $f(x)$ sedemikian sehingga $deg\ f(X) < t$ dan $f(j) = share_j(s)$ untuk $j=1,2,..t$.
   b. Pembangkitan *secret S* yang merupakan $f(0)$.



Gambar 3.1 Kurva *Shamir Secret Sharing Scheme*

Diberikan sejumlah *t* pasang (*i,f(i)*) dengan nilai *i* yang berbeda-beda (gambar 3.1), maka akan terdapat persamaan polinomial yang unik *f(x)* dengan derajat t – 1, yang melewati semua titik-titik tersebut. Polinomial ini dapat dihitung dari pasangan titik-titik yang telah diberikan dengan menggunakan persamaan P 3.4 sebagai berikut :

$$a(x) = \sum_{i=1}^{m+1} y_i L_i(x) \qquad \text{P 3.4}$$

dengan $L_i(x)$ adalah polinomial *Lagrange* (P 3.5), yaitu :

$$L_i(x) = \frac{\Pi_{j \neq i}(x - x_j)}{\Pi_{j \neq i}(x_i - x_j)} \qquad \text{P 3.5}$$

Ini adalah sistem persamaan linear dengan dengan nilai *t* yang tidak diketahui, dengan sejumlah *t* persamaan. Sistem persamaan linear ini mempunyai solusi yang unik. Untuk menyelesaikan sistem persamaan ini, dapat digunakan eliminasi *Gauss*.

Metode penghitungan keamanan berhubungan dengan nilai ambang. Misalkan dengan nilai ambang *n,* berhasil dikumpulkan *shares* sejumlah *n*-1. Hal ini berpadanan dengan suatu sistem persamaan linear dengan *n* peubah dan sejumlah *n*-1 persamaan linear. Sistem persamaan linear ini akan mempunyai solusi yang tak terhingga. Sebagai contoh sederhana, misalkan nilai ambang adalah 3, dan jumlah *shares* yang berhasil dikumpulkan adalah 2. Maka akan terbentuk sistem persamaan linear dengan 3 peubah dan 2 persamaan linear sebagai berikut:

$$\begin{aligned} 4x_1 - x_2 + 3x_3 &= -1 \\ 3x_1 + x_2 + 9x_3 &= -4 \end{aligned} \qquad \text{P 3.6}$$

Misalkan dilakukan substitusi persamaan pertama pada persamaan kedua, maka akan diperoleh persamaan sebagai berikut:

$$-9x_1 - 4x_2 = -1 \qquad \text{P 3.7}$$

Persamaan P 3.7 adalah sebuah persamaan garis lurus dengan panjang tak terhingga, yang merupakan solusi dari sistem persamaan linear P 3.6. Hal ini akan berlaku juga untuk sebarang nilai ambang. Solusi ini tidak menggambarkan informasi apapun tentang *secret*. Jika dilakukan usaha untuk mencoba-coba, maka harus dilakukan untuk semua titik sepanjang persamaan garis P 3.7 tersebut.

4. Pengembangan *Shamir Secret Sharing* (*Shamir Secret Sharing Advanced*)

Pengembangan metode *Shamir secret sharing* dilakukan dengan mengelompokkan penerima *shares* (*participants*) menjadi beberapa kelompok. Pembagian *secret* pada metode ini dilakukan dua *level*, yaitu *level clusters* dan *level participants*. Pembagian *level clusters* dapat dilakukan dengan dua metode, yaitu pembagian linear dan pembagian *(n,n) threshold scheme*. Untuk pembagian *level participant*, digunakan *Sharmir secret sharing* dengan *(k,n) threshold scheme*. Variabel jumlah *clusters* tidak rahasia. Variabel jumlah *participants* dan jumlah minimal *participants* rahasia pada tingkat *clusters*, tapi tidak rahasia pada tingkat *participants* dalam suatu *cluster*. *Shares* yang dibangkitkan adalah rahasia pada tingkat *participants*.

Dua *protocol* utama pada skema ini mirip dengan *protocol-protocol* pada S*hamir secret sharing*, dengan beberapa langkah tambahan, sebagai berikut:

1. *Sharing Protocol*
   a. *Dealer* membagi secret *S* menjadi sejumlah *clusters*.
      $S \rightarrow (S_1,S_2,S_3,\ldots,S_n)$
   b. Untuk setiap S_n, *dealer* membangkitkan persamaan polinomial secara random $f(x_n)$ dengan derajat *t–1*, dengan persamaan P 4.1. Variabel *t* merupakan nilai ambang (*threshold value*).

$$f(x_n) = a_0 + a_1 x_n + \ldots + a_{t-1} x_n^{t-1} \qquad \text{P 4.1}$$

      Polinomial ini merupakan persamaan tidak tak-terhingga, dengan a_0 adalah S_n dan koefisien-koefisien lainnya dipilih secara random.
   c. Untuk setiap *cluster, dealer* memilih *n* titik random yang berbeda-beda, dengan $x_j \neq 0$, dan membagikannya secara rahasia kepada setiap *participants.* Untuk *participant* P_j akan mendapatkan $share_j(s) = (x_j, f(x_j))$ untuk $j = 1, 2, \ldots, n$.
2. *Reconstruction Protocol*
   a. Interpolasi persamaan polinom untuk mendapatkan persamaan polinomial yang unik $f(x_n)$ sedemikian sehingga $deg\ f(X) < t$ dan $f(j) = share_j(s)$ untuk $j=1,2,..t.$

b. Untuk setiap *cluster*, dibangkitan S_n yang merupakan *f(0)* untuk fungsi $f(x_n)$ pada *cluster* tersebut.
c. Dari $S_1,S_2,S_3, \ldots, S_n$ pada langkah b, dibangkitan *S* yang merupakan *f(0)* untuk fungsi *f(x)*, salah satu contohnya dengan menggunakan eliminasi *Gauss*.

5. Contoh Metode *Shamir Secret Sharing Advanced*

Berikut ini adalah contoh sederhana metode *Shamir secret sharing advanced*.

Sebuah kunci (M = 11) ingin dibagi menjadi lima *shares* sedemikian sehingga dibutuhkan minimal tiga dari lima *shares* untuk membentuk kunci awal. Pada contoh disini, pembagian pertama menggunakan (*n,n*) *threshold shamir secret sharing*, yaitu kunci M=11 dibagi menjadi tiga bagian sedemikian sehingga dibutuhkan sejumlah tiga *shares* untuk membentuk kunci awal. Tiap *shares* yang terbentuk disini didistribusikan kepada tiap *cluster*. *Shares* ini akan menjadi kunci untuk dibagi lagi di tingkat *clusters*.

$f(x) = (11 + 8x + 7x^2) \bmod 13$

tiga *shares* yang terbentuk pada pembagian tingkat *clusters*:

$c_1 = f(1) = (7 + 8 + 11) \bmod 13 = 0$
$c_2 = f(2) = (28 + 16 + 11) \bmod 13 = 3$
$c_3 = f(3) = (63 + 24 + 11) \bmod 13 = 7$

Shares ini akan menjadi kunci untuk dibagi lagi di tiap *clusters* menggunakan skema (*k,n*) *threshold shamir secret sharing*. Rule (*k,n*) spesifik untuk tiap *clusters*.

Untuk *cluster* pertama, misalnya menggunakan skema (3,5) *shamir secret sharing* dengan kunci M = 0, sebagai berikut :

$f(x) = (0 + 2x + 3x^2) \bmod 5$

lima *shares* yang terbentuk pada pembagian tingkat *participants*:

$k_1 = f(1) = (2 + 3 + 0) \bmod 5 = 0$
$k_2 = f(2) = (4 + 12 + 0) \bmod 5 = 1$
$k_3 = f(3) = (6 + 27 + 0) \bmod 5 = 3$
$k_4 = f(4) = (8 + 48 + 0) \bmod 5 = 1$
$k_5 = f(5) = (10 + 75 + 0) \bmod 5 = 0$

Untuk *cluster* kedua, misalnya menggunakan skema (2,3) *shamir secret sharing* dengan kunci M = 3, sebagai berikut :

$f(x) = (3 + 4x) \bmod 11$

tiga *shares* yang terbentuk pada pembagian tingkat *participants*:

$k_1 = f(1) = (4 + 3) \bmod 11 = 7$
$k_2 = f(2) = (8 + 3) \bmod 11 = 0$
$k_3 = f(3) = (12 + 3) \bmod 11 = 4$

Untuk *cluster* ketiga, misalnya menggunakan skema (2,2) *shamir secret sharing* dengan kunci M = 7, sebagai berikut :

$f(x) = (7 + 1x) \bmod 17$

dua *shares* yang terbentuk pada pembagian tingkat *participants*:

$k_1 = f(1) = (1 + 7) \bmod 17 = 8$
$k_2 = f(2) = (2 + 7) \bmod 17 = 9$

Untuk membangkitkan kunci awal, diperlukan 2 tingkat interpolasi sistem persamaan linear. Interpolasi pertama pada tingkat *participants*. Solusi pada interpolasi tingkat *participants* akan digunakan untuk interpolasi sistem persamaan linear tingkat *clusters*.

Dari *cluster* pertama, misalnya digunakan *shares* k_2, k_3 dan k_4. Sistem persamaan linear yang terbentuk sebagai berikut :

$k_2 = f(2) = (a * (2)^2 + b*2 + M) \bmod 5 = 1$
$k_3 = f(3) = (a * (3)^2 + b*3 + M) \bmod 5 = 3$
$k_4 = f(4) = (a * (4)^2 + b*4 + M) \bmod 5 = 1$

Dengan melakukan interpolasi persamaan diatas, maka akan didapatkan solusi $a = 3$, $b = 2$, $M = 0$.

Dari *cluster* kedua, misalnya digunakan *shares* k_1, dan k_2. Sistem persamaan linear yang terbentuk sebagai berikut :

$k_1 = f(1) = (a * (1) + M) \bmod 11 = 7$
$k_2 = f(2) = (a * (2) + M) \bmod 11 = 0$

Dengan melakukan interpolasi persamaan diatas, maka akan didapatkan solusi $a = 7$, $b = 4$, $M = 3$.

Dari *cluster* ketiga, harus digunakan semua *shares*, yaitu *shares* k_1, dan k_2. Sistem persamaan linear yang terbentuk sebagai berikut :

$k_1 = f(1) = (a * (1) + M) \bmod 17 = 8$
$k_2 = f(2) = (a * (2) + M) \bmod 17 = 9$

Dengan melakukan interpolasi persamaan diatas, maka akan didapatkan solusi $a = 5$, $b = 1$, $M = 7$.

Dengan demikian, didapatkan *shares* untuk pembangkitan kunci awal, yaitu 0, 3, dan 7. Untuk membentuk kembali kunci, digunakan ketiga *shares* ini, yaitu c_1, c_2, dan c_3. Sistem persamaan linear yang terbentuk :

$c_1 = f(1) = (a * (1)^2 + b*1 + M) \bmod 13 = 0$
$c_2 = f(2) = (a * (2)^2 + b*2 + M) \bmod 13 = 3$
$c_3 = f(3) = (a * (3)^2 + b*3 + M) \bmod 13 = 7$

Dengan melakukan interpolasi persamaan diatas, maka akan didapatkan solusi $a = 7$, $b = 8$, $M = 11$, dengan M=11 adalah kunci awal.

6. Kesimpulan

1. Metode *Shamir secret sharing advanced* merupakan solusi yang dapat digunakan untuk permasalahan kepemilikan bersama suatu *secret* dengan membagi *secret* tersebut kepada beberapa *participants*. Metode *Shamir secret sharing advanced* dengan penerapan *clustering participants* akan menghasilkan struktur akses yang bervariasi. Dengan perancangan struktur akses, maka akan didapatkan pendistribusian kekuatan *shares* yang diinginkan.
2. Metode *Shamir secret sharing advanced* merupakan metode yang aman untuk pembagian *secret*. *Secret* tidak akan dapat dibangkitkan jika nilai ambang tidak terpenuhi. *Secret* juga tidak akan dapat dibangkitkan jika *shares* telah mengalami suatu manipulasi.

3. Keamanan *secret* akan berbanding lurus dengan nilai ambang. Semakin besar nilai ambang, maka akan semakin aman *secret* tersebut karena dengan semakin besar nilai ambang maka akan dibutuhkan semakin banyak *shares* untuk membentuk kembali *secret.* Dan semakin besar nilai ambang maka sistem persamaan linear yang dibangkitkan akan mempunyai variabel yang semakin banyak sehingga akan semakin sulit untuk menemukan solusi persamaan.

7. Daftar Pustaka

1. Canny,John, "*Secret Sharing and Threshold Description*", Lecture Notes CS174, Berkeley University,2002, http://www.cs.berkeley.edu/~jordan/courses/174-spring02/notes/note17.pdf
   Tanggal Akses : 15 Maret 2005
2. Micciancio, Daniele, "*Advanced cryptography*", Lecture Notes, University of California, San Diego,2002, http://www.cse.ucsd.edu/classes/fa02/cse208/lec12.html
   Tanggal Akses : 15 Maret 2005
3. Menezes,A, Van Oorschot,P, Vanstone,S, "*Handbook of Applied Cryptography*", CRC Press, 1997.
4. Stinson, Douglas, "*Cryptography Theory And Practice*", CRC Press, ISBN: 0849385210, 1995.
5. Shamir,Adi, "*How to Share a Secret*", Communication of the ACM, Massachusetts Institute of Technology, 1979, http://szabo.best.vwh.net/secret.html
   Tanggal Akses : 15 Maret 2005