Otentikasi Aplikasi *Group Chat* dengan memanfaatkan *Secret Sharing Scheme*

Yoga Adrian Saputra (13513030)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Infomatika, Institut Teknologi Bandung

Jalan Ganesha 10-12, Bandung 40132, Indonesia

yoga.adrian@students.itb.ac.id

***Abstract*— Otentikasi merupakan sebuah konsep yang dipikirkan dalam ilmu kriptografi. Secara singkat, otentikasi merupakan verifikasi yang membuktikan bahwa orang tersebut adalah orang yang berhak. Salah satu ilmu kriptografi dalam menyelesaikan masalah otentikasi adalah *secret sharing scheme*. Otentikasi dalam *group chat* digunakan agar informasi yang mengalir, benar tertuju pada anggota dalam *group chat* tersebut**

Keywords*— Group chat, Kriptografi, Otentikasi, Secret sharing scheme

I. PENDAHULUAN

Pada masa ini teknologi komputer berkembang dengan sangat cepat. Apalagi pada era informasi ini, perkembangan teknologi komputer berkembang ke arah informasi. Analisis informasi memberikan banyak sekali manfaat yang dapat mendukung kehidupan kita. Seperti pada bidang bisnis, otomatisasi, dan masih banyak lagi. Bahkan pada era ini, informasi sangat mudah ditemukan oleh masyarakat.

Namun terdapat banyak pula dampak negatif yang didapat dari perkembangan bidang informasi dalam teknologi ini. Untuk orang yang cukup mengerti di bidang teknologi, pengambilan informasi rahasia atau milik orang lain bisa dilakukan dengan mudah. Mengambil informasi milik orang lain berarti privasi orang tersebut berkurang. Privasi merupakan hak yang dimiliki oleh setiap orang. Dengan mengambil informasi milik orang lain, banyak kejahatan yang bisa dilakukan kepada orang tersebut. Kejahatan tersebut bisa berguna untuk keuntungan pengambil informasi tersebut atau memang dimaksudkan untuk merugikan orang tersebut.

Komunikasi merupakan hal yang dilakukan setiap hari oleh setiap orang. Komunikasi yang dilakukan bisa berupa komunikasi antara 2 orang ataupun lebih. Dalam perkembangan teknologi masa ini, komunikasi juga bisa dilakukan dalam dunia dijital. Bahkan mungkin komunikasi yang dilakukan dalam dunia dijital jauh lebih sering daripada komunikasi langsung (tatap muka). Informasi-informasi penting juga banyak yang disalurkan lewat komunikasi dalam dunia dijital.

Komunikasi dalam dunia dijital inilah yang bisa dimanipulasi oleh orang yang mengerti seluk beluk sistem komunikasi tersebut. Komunikasi tersebut dimanipulasi agar orang tersebut bisa mendapatkan informasi yang seharusnya ia tidak dapatkan. Banyak cara manipulasi yang bisa dilakukan agar orang tersebut bisa mengetahui informasi yang mengalir didalam jalur komunikasi tersebut. Ini merupakan salah satu dampak negatif dari perkembangan teknologi dalam bidang informasi jika tidak diatasi dengan baik.

Kriptogafi merupakan salah satu bidang ilmu pengetahuan informatika yang bergerak untuk keamanan informasi. Banyak sekali konsep yang ditemukan agar keamanan informasi terjaga dengan baik. Hanya orang yang berhak mengetahui informasi tersebut yang mendapatkan informasi tersebut. Banyak sekali konsep yang harus dipikirkan agar informasi yang dikirimkan benar benar hanya sampai kepada orang yang dituju. Salah satu konsep yang dipikirkan dalam kriptografi adalah otentikasi. Otentikasi adalah proses verifikasi yang membuktikan orang yang terlibat dalam komunikasi atau pertukaran informasi adalah orang yang benar. Hal ini bisa dilakukan dengan menerapkan pemberian kunci kepada orang yang berhak. Dengan memiliki kunci, orang tersebut dapat melakukan komunikasi atau pertukaran informasi dengan orang lain yang juga memiliki kunci.

Dalam makalah ini, akan dilakukan percobaan implementasi dan analisis *Secret Sharing Scheme* kedalam bentuk komunikasi dijital antara banyak orang (*Group chat*). Dengan menggunakan *Secret Sharing Scheme* diharapkan orang yang menerima informasi dalam *group chat* adalah orang yang benar merupakan anggota dalam *group chat*

tersebut. Sehingga privasi setiap orang dalam *group chat* tersebut dapat terjaga dengan baik.

II. DASAR TEORI

A. *Secret Sharing Scheme*

Secret Sharing Scheme adalah salah satu cara untuk membagi sebuah nilai Integer ke beberapa orang. Nilai Integer tersebut biasanya berupa sebuah informasi penting. Nilai Integer tersebut bisa berupa pesan yang di bentuk sebuah angka atau berupa kode rahasia. Angkat tersebut akan dibagi menjadi beberapa bagian dan dibagikan ke beberapa orang. Untuk membuat menjadi angka atau kode semula, beberapa bagian hasil pembagian angka tersebut harus disatukan lewat sebuah algoritma.

Berikut adalah langkah-langkah skema yang digunakan untuk membagi sebuah angka M ke w orang dengan minimal t orang.

- Pilih sebuah angka p yang pasti lebih besar dari semua kemungkinan angka M.
- Ambil bilangan acak sebanyak t-1 angka dan simpan dalam bentuk polinom bersama M

$$s(x) \equiv M + R_1X^1 + \ldots.. + R_{t-1}X^{t-1} \pmod p$$

Rn : Bilangan acak ke-n

Bentuk polinom ini harus dirahasiakan dan tidak diketahui oleh calon pemegang bagian (*share*).

- Untuk setiap partisipan , akan diberikan nilai x yang berbeda beda. Lalu dari x tersebut, s(x) akan dihitung dan diberikan bersama dengan x ke partisipan tersebut

Nilai x dan y inilah bagian yang didapatkan oleh masing masing partisipan. Sebanyak t bagian dari t partisipan akan digabung untuk mengembalikan angka atau kode M yang semula.

Ide dari skema ini adalah untuk memecahkan permasalahan persamaan matematika oleh t-1 polinom. Dengan memilik x dan y sebanyak t, persamaan matematika tersebut bisa dipecahkan.

Hal tersebut berlaku untuk mengembalikan beberapa bagian menjadi angka atau kode M semula. Ada beberapa cara untuk mendapatkan angka atau kode M semula yaitu: Eliminasi gauss atau Interpolasi Lagrage. Dalam pengimplementasiannya, penulis menggunakan interpolasi Lagrage untuk menyelesaikan persamaan matematika dalam *Secret Sharing Scheme*

B. *Lagrage Interpollation*

Lagrage Interpollation adalah metode untuk membentuk Persamaan Lagrage dari beberapa titik yang diketahui. Persamaan lagrage tersebut bisa dibentuk dengan cara memasukkan titik tersebut kedalam rumus yang disebarluaskan oleh Lagrage: yaitu

$$P(x) = \sum_{j=1}^{n} P_j(x),$$

Dimana

$$P_j(x) = y_j \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}.$$

Jika dituliskan lebih rinci

$$P(x) = \frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_n)} y_1 + \frac{(x-x_1)(x-x_3)\cdots(x-x_n)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_n)} y_2 + \cdots + \frac{(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})} y_n.$$

Gambar 1. Rumus pembentuk persamaan Lagrage

Sumber : http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html

Diakses pada tanggal 18 Mei 2016 Pukul 22.22

Namun karena digunakan *Secret Sharing* Scheme, kalkulasi selalu dilakukan pada bilangan bulat dengan modulus p (angka yang diambil dalam langkah awal *Secret Sharing Scheme*). Maka untuk mendapatkan hasil yang benar, kalkulasi dilakukan menurut dengan teorema aritmatika modulo.

Karena pada persamaan polinom yang dibuat dari *Secret Sharing Scheme* adalah

$$s(x) \equiv M + R_1X^1 + \ldots.. + R_{t-1}X^{t-1} \pmod p$$

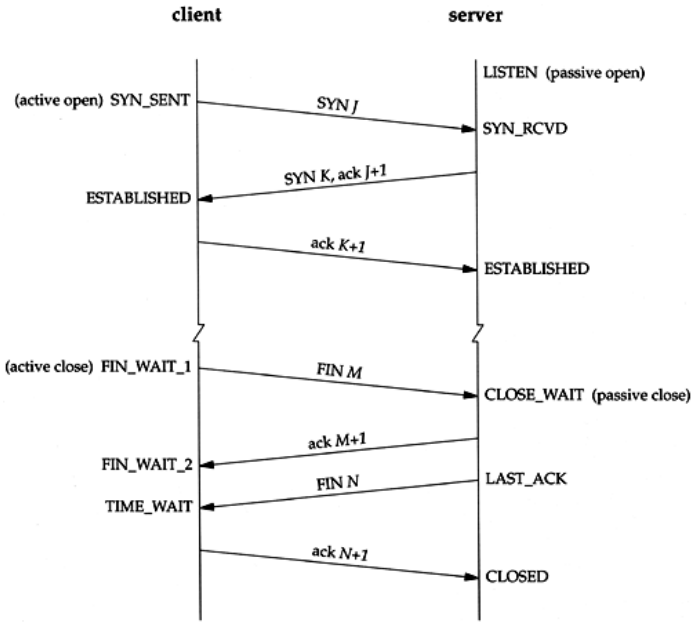
Diketahui bahwa angka atau kode M yang disimpan adalah koefisien dalam x derajat 0, Nilai M dapat diketahui ketika nilai x diisi 0 setelah mendapatkan persamaan yang dimaksud.

Lalu akan dicek nilai M yang didapat dari interpolasi Lagrage *share-share* partisipan yang ingin masuk dengan nilai M yang didapatkan dari bilangan acak yang disimpan di awal *Secret Sharing Scheme*. Jika benar, maka partisipan tersebut benar-benar partisipan yang berhak. Jika salah maka ada partisipan yang tidak seharusnya ada disana.

C. TCP(Transmission Cotrol Protocol)

Transmission Control Protocol adalah protokol yang digunakan untuk membentuk dan menjalankan komunikasi dijital lewat jaringan melalui sebuah aplikasi. TCP bersama dengan standar protokol lainnya (IP), mengatur bagaimana cara mengirim, menerima paket-paket, mengatur alur paket, serta deteksi error bila terjadi pengiriman aneh antara 1 komputer dengan komputer lain.

TCP merupakan *connection-oriented protocol*. Sehingga untuk melakukan komunikasi, 2 belah pihak harus setuju untuk membuat koneksi. Koneksi itu digunakan untuk mengirimkan pesan. Koneksi itu akan dijaga hingga proses komunikasi selesai.



Gambar 2 Diagram pembentukan dan pehancuran koneksi TCP

Sumber :
http://flylib.com/books/3/223/1/html/2/files/18fig13.gif

Diakses pada tanggal 19 Mei 2016 Pukul 02.00

Keuntungan menggunakan TCP adalah

- Pengiriman paket yang *reliable*
- Pengiriman paket sudah teratur
- Terdapat pengecekan error terhadap pengiriman paket.

III. IMPLEMENTASI

Secara umum cara implementasi yang penulis gunakan untuk mengembangkan aplikasi ini adalah sebagai berikut

- Bahasa yang digunakan oleh penulis adalah Bahasa Java.
- Terdapat 2 program utama. Program utama tersebut adalah *Server* dan *Client*.
- 2 Program tersebut dibuat agar dapat berkomunikasi satu sama lain (antara *server* dengan *client*). Dalam 1 *server* bisa terdapat beberapa *client* yang terhubung. Komunikasi antara *server* dan *client* menggunakan Transmission Control Protocol (TCP).
- Alamat *server* akan muncul ketika program *server* berjalan. Agar terhubung, *client* hanya perlu memasukkan alamat *server* dalam alamat yang diminta saat program *client* dijalankan.
- Program *server* dan *client* dibuat agar alur bersifat command-based. Artinya *client* akan menuliskan suatu perintah kepada *server*. *Server* akan memproses sesuai perintah yang diberikan oleh *client* dan kondisi *server* saat itu.
- Beberapa perintah yang didefinisikan adalah
    - create : Berguna untuk membuat group chat baru. Seluruh *client* yang sekarang terkoneksi dengan *Server* akan diberikan bagian (*share*) sesuai *Secret Sharing* Scheme yang sudah didefinisikan. Angka atau kode M dalam *Secret Sharing* Scheme akan dibentuk secara acak dan berguna sebagai kunci dalam layanan group chat. Kunci tersebut akan digunakan untuk menentukan apakah *client* yang terhubung saat ini adalah *client* yang benar dan berhak untuk mengirim dan menerima informasi dalam layanan group chat. *Share* yang masing-masing diterima oleh *client* akan disimpan dalam suatu file yang namanya sudah definisikan oleh *client* dalam command “save”. Angka M juga akan disimpan oleh *Server* dalam suatu file.
    - save <nama_file> : berguna untuk menyimpan *share* yang didapatkan nantinya dari *server* kedalam file eksternal <nama_file>
    - send <nama_file> : berguna untuk mengirimkan *share* sesuai yang tertulis dalam file <nama_file> ke *server*. Proses ini merupakan proses otentikasi *client* yang membuktikan apakah *client* tersebut benar-benar berhak. Jika ia memang berhak, *client* tersebut pasti memiliki nilai *share* yang

dibagikan oleh *server* ketika commang "create".

- chat <pesan> : berguna untuk mengirimkan pesan ke semua *client* yang terhubung. Layanan ini berfungsi setelah dilakukan pengecekan terhadap *share* yang dimiliki oleh setiap *client.*

- Setelah semua *client* melakukan command "send", *server* akan mengecek *share* yang didapat dengan interpolasi Lagrage. Jika angka M yang didapat dari interpolasi Lagrage dan angka M yang disimpan dalam file sama. Maka semua *client* tersebut benar merupakan *client* yang berhak melakukan komunikasi. Jika berbeda, maka ada *client* yang tidak berhak melakukan komunikasi.
- Setelah dilakukan pengecekan *share* dan benar, maka para *client* akan diperbolehkan untuk berkomunikasi satu sama lain. Proses pengirimannya adalah dikirimkan ke *server* terlebih dahulu, lalu akan disebarkan ke semua *client* dengan command chat <pesan>.
- Dalam 1 *server* , hanya diberikan layanan 1 *group chat*. Jika *client* melakukan command "create", maka group chat sebelumnya akan dihapus.
- Untuk mengimplementasi angka, digunakan kombinasi long dengan BigInteger. BigInteger digunakan pada perhitungan Interpolasi Lagrage. Sedangkan long di tempat lainnya.

IV. EKSPERIMAN DAN PEMBAHASAN HASIL

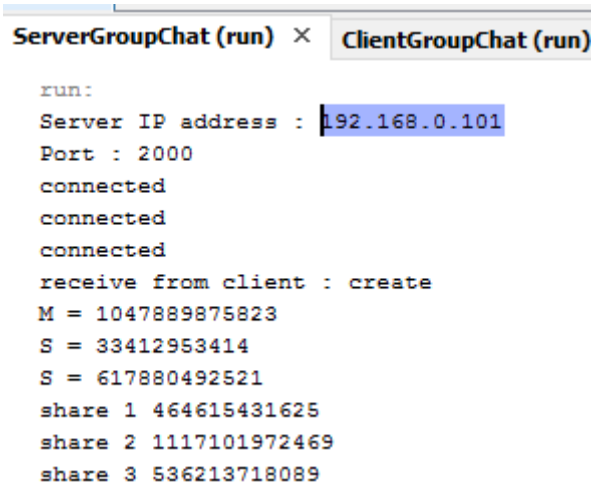
Berikut adalah pengujian yang dilakukan untuk mengukur kualitas keamanan dalam aplikasi

A. *Kondisi Pengujian*

p untuk *secret sharing* schemes = 1234567890133

jumlah *client* = 3

Terminal *Server*



Gambar 3 Screenshot tampilan *server* setelah Create

Terminal *Client* 1

```
Input server IP hostname : 192.168.0.101
save share1.txt
create
Receive from server : share 1 464615431625
```

Isi *share*1.txt pada *Client* 1

```
1 464615431625
```

Isi *secret*.txt pada *server*

```
1047889875823
```

Isi pada *client* 2 dan 3 sama seperti *client* 1, hanya berbeda *share*.

B. *Pengujian Kebenaran Algoritma*

Terminal *Client* 1

```
send share1.txt
```

Isi pada *client* 2 dan 3 sama seperti *client* 1, hanya yang dikirim adalah *share*2.txt dan *share*3.txt

Terminal *Server*

```
receive from client : send 1 464615431625
receive from client : send 2 1117101972469
receive from client : send 3 536213718089
ini m dari interpolasi 1047889875823
ini m awal 1047889875823
permitted
```

Terminal *client* 1

```
chat halooooooo
Receive from server : 0: halooooooo
```

Terminal *client* 2 & 3

```
Receive from server : 0: halooooooo
```

C. *Pengujian jumlah client kurang dari saat create(2 client)*

Pengujian 1

Terminal *client* 1

```
Input server IP hostname : 192.168.0.101
send share1.txt
```

Terminal *client* 2

```
Input server IP hostname : 192.168.0.101
send share2.txt
```

Terminal *Server*

```
Server IP address : 192.168.0.101
Port : 2000
connected
connected
receive from client : send 1 464615431625
receive from client : send 2 1117101972469
ini m dari interpolasi 1046696780914
ini m awal 1047889875823
not permitted
```

Pengujian 2

Terminal *client* 1

```
Input server IP hostname : 192.168.0.101
send share1.txt
```

Terminal *client* 2

```
Input server IP hostname : 192.168.0.101
send share3.txt
```

Terminal *Server*

```
Server IP address : 192.168.0.101
Port : 2000
connected
connected
receive from client : send 1 464615431625
receive from client : send 3 536213718089
ini m dari interpolasi 428816288393
ini m awal 1047889875823
not permitted
```

D. Pengujian jumlah client lebih dari saat create (1 share rekayasa)

Client 1 , 2 , dan 3 mengisi *share* milik mereka yang benar

Terminal *client* 4

```
Input server IP hostname : 192.168.0.101
send sharerekayasa.txt
```

Isi *share*rekayasa.txt

```
4 736215718089
```

Terminal *Server*

```
Server IP address : 192.168.0.101
Port : 2000
connected
connected
connected
connected
receive from client : send 1 464615431625
receive from client : send 2 1117101972469
receive from client : send 3 536213718089
receive from client : send 4 736215718089
ini m dari interpolasi 268192716352
ini m awal 1047889875823
not permitted
```

E. Pengujian salah satu share yang diberikan salah

Pengujian 1.

Client 2 & 1 mengisi *share* miliknya yang benar.

Isi *share*3.txt awal

```
3 536213718089
```

Isi *share*3.txt yang y nya dirubah

```
3 536213718189
```

Terminal *server*

```
Server IP address : 192.168.0.101
Port : 2000
connected
connected
connected
receive from client : send 1 464615431625
receive from client : send 2 1117101972469
receive from client : send 3 536213718189
ini m dari interpolasi 1047889875923
ini m awal 1047889875823
inifalse
```

Pengujian 2

Client 2 & 1 mengisi *share* miliknya yang benar.

Isi *share*3.txt awal

```
3 536213718089
```

Isi *share*3.txt yang x nya dirubah

```
23 536213718089
```

Terminal *server*

```
Server IP address : 192.168.0.101
Port : 2000
connected
connected
connected
receive from client : send 23 536213718089
receive from client : send 2 1117101972469
receive from client : send 1 464615431625
ini m dari interpolasi 386286783914
ini m awal 1047889875823
not permitted
```

Pengujian 3

Client 2 & 1 mengisi *share* miliknya yang benar.

Isi *share*3.txt awal

```
3 536213718089
```

Isi *share*3.txt yang x dan y nya dirubah

```
23 536213718189
```

Terminal *server*

```
Server IP address : 192.168.0.101
Port : 2000
connected
connected
connected
receive from client : send 23 536213718189
receive from client : send 2 1117101972469
receive from client : send 1 464615431625
ini m dari interpolasi 140442095836
```

ini m awal 1047889875823 not permitted

F. Pembahasan Hasil

Dari hasil pengujian, aplikasi dapat berjalan sesuai harapan ketika *share* yang diberikan benar dari semua *client* yang terdaftar saat command "create" diberikan.

Dari hasil pengujian C(jumlah *client* kurang dari saat command "create") dapat dilihat ketika menggunakan *share* 1 dan *share* 2, nilai M hanya berbeda sedikit dari sebelumnya. Sedangkan pada penggunaan *share* 1 dan 3, nilai M sedikit lebih acak. Hal ini juga dikuatkan oleh pengujian E(Pengujian salah satu *share* salah). Dari pengujian E, terlihat bahwa ketika mengganti y saat x nya kecil (3), nilai M yang terbentuk hanya berbeda sedikit, bahkan sesuai dengan perubahan nilai *share* pada *share* 3. Sedangkan ketika mengubah nilai x menjadi 23 dan nilai y diubah sedikit, nilai M jauh lebih acak. Hal ini terjadi karena dalam metode interpolasi Lagrega, penentuan nilai M didasarkan dari persamaan polynomial yang terbentuk. Ketika kita menggunakan nilai x yang relatif dekat, lalu kita mengganti nilai y nya maka persamaan polynomial yang terbentuk tidak terlalu berbeda. Hal ini akan berbeda jika kita mengambil jarak x yang cukup berbeda.

Dari hasil pengujian D (jumlah *client* lebih banyak dibanding saat command "create"), nilai M cukup acak ketika diambil nilai yang acak.

G. Analisis Keamanan

Dari pembahasan hasil, terlihat bahwa nilai M yang salah, akan jauh lebih sensitive bila menggunakan nilai x yang jaraknya relatif jauh antara 1 *share* dengan *share* yang lain. Dengan menambahkan sensifitas nilai M, maka semakin kecil pula kemungkinan terjadinya bocornya informasi kepada pihak yang seharusnya tidak mendapatkan informasi tersebut.

Dari berbagai macam pengujian, juga didapatkan program hanya berjalan benar dengan skenario yang benar. Selain daripada skenario tersebut, maka program tidak akan mengijinkan untuk melakukan komunikasi. Sehingga dalam masalah fungsional, aplikasi ini mampu menawarkan otentikasi yang cukup.

Keamanan dari pengiriman juga cukup karena protokol yang digunakan untuk mengirimkan pesan maupun algoritmanya menggunakan *transmission control protocol*.

V. KESIMPULAN DAN SARAN

Aplikasi ini dapat mengatasi masalah otentikasi dalam komunikasi dijital antar banyak orang. Walaupun sebenarnya banyak aplikasi chatting yang memberikan layanan *group chat*, mungkin dengan mencatat daftar grup dalam database yang memiliki otentikasi yang bagus, namun aplikasi ini menawarkan aplikasi *group chat* dalam kondisi yang *simple*. Yaitu hanya perlu mempunyai file yang berisi *share* yang dimiliki.

Untuk saran pengembang berikutnya adalah menambahkan fungsional yang menabah nilai kualitas aplikasi atau mengubah fungsional-fungsional yang masih janggal dalam aplikasi saat ini. Seperti hanya menyediakan 1 layanan group chat dan dapat dihapus jika melakukan command "create" lagi.

REFERENSI

[1] Jeffreys, H. and Jeffreys, B. S. "Lagrange's Interpolation Formula." §9.011 in Methods of Mathematical Physics, 3rd ed. Cambridge, England: Cambridge University Press, p. 260, 1988J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] Hazewinkel, Michiel, ed. (2001), "Lagrange interpolation formula",Encyclopedia of Mathematics, Springer, ISBN 978-1-55608-010-4

[3] http://searchnetworking.techtarget.com/definition/TCP

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