DESAIN DAN IMPLEMENTASI PROTOKOL KRIPTOGRAFI UNTUK APLIKASI *SECURE CHAT* PADA *MULTIPLATFORM* SISTEM OPERASI

Faizal Achmad
Lembaga Sandi Negara
e-mail : faizal.achmad@lemsaneg.go.id

Abstrak
Permasalahan yang terkait dengan keamanan informasi adalah masalah kerahasiaan, otentikasi dan keutuhan data serta nir-penyangkalan. Salah satu teknik yang dapat digunakan untuk melindungi informasi adalah dengan menggunakan kriptografi. Mekanisme kriptografi digunakan untuk membangun protokol kriptografi yang bertujuan untuk mencapai fungsi yang terkait keamanan informasi. Pada penelitian ini penulis membuat desain dan implementasi protokol kriptografi pada aplikasi Secure Chat yang dapat dijalankan pada multiplatform sistem operasi. Protokol kriptografi hasil perancangan telah diimplementasikan pada aplikasi Secure Chat untuk mengamankan komunikasi chat antar pengguna dengan menggunakan bahasa pemrograman Java.

Kata Kunci : *Keamanan Informasi, Protokol Kriptografi, Secure Chat, Multiplatform, Sistem Operasi.*

1. PENDAHULUAN

1.1 Latar Belakang Masalah

Semakin berkembangnya ilmu pengetahuan, serta terintegrasinya teknologi informasi, komputer dan telekomunikasi yang semakin global, memungkinkan setiap individu atau kelompok untuk dapat mengakses suatu data informasi tanpa kenal batas. Ancaman yang kerap terjadi terhadap informasi pada suatu protokol komunikasi antara lain pengubahan, penyadapan, dan pemalsuan. Salah satu teknik pengamanan terhadap suatu informasi rahasia adalah dengan kriptografi. Untuk membatasi agar suatu data informasi yang berklasifikasi rahasia hanya dapat diakses oleh pihak-pihak yang berkepentingan, maka perlu dibuat suatu protokol komunikasi yang memanfaatkan beberapa mekanisme algoritma kriptografi atau yang biasa disebut dengan protokol kriptografi. Secara umum kriptografi adalah ilmu untuk menjaga kerahasiaan data melalui penggunaan teknik-teknik tertentu, tujuannya untuk mengamankan komunikasi penting atau rahasia antara pihak yang saling berkepentingan.

Pada penelitian ini penulis akan membuat desain protokol kriptografi dan mengimplementasikannya pada aplikasi Secure Chat sebagai solusi komunikasi rahasia, antara user yang menggunakan multiplatform sistem operasi.

1.2 Tujuan Penulisan

Tujuan penulisan adalah desain dan implementasi protokol kriptografi untuk mengamankan data informasi rahasia yang berbentuk komunikasi *chat,* serta pengamanan *password Login*. Hasil implementasi ini diharapkan dapat dijalankan pada *multiplatform* sistem operasi dan dimanfaatkan untuk mengamankan komunikasi *chat* yang akan ditransmisikan melalui jalur komunikasi publik.

1.3 Perumusan Masalah

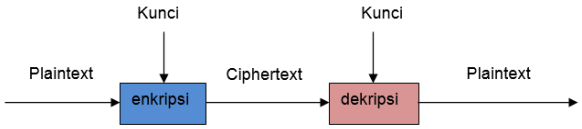
Dalam aplikasi keamanan suatu data terdapat beberapa hal permasalahan sebagai berikut :

1) Keamanan suatu data yang ditransmisikan melalui jalur komunikasi rawan terhadap pengubahan, penyadapan dan pemalsuan.
2) Untuk pengamanan suatu data informasi rahasia dibutuhkan suatu algoritma kriptografi yang sudah teruji tingkat keamanannya.
3) Aplikasi *Secure Chat* diharapkan dapat berjalan pada *multiplatform* sistem operasi.

2. LANDASAN TEORI

2.1 Definisi Kriptografi

Berikut merupakan istilah-istilah yang terdapat dalam kriptografi (Schneier. 1996) :
Enkripsi adalah proses menyamarkan suatu pesan sebagai cara untuk menyembunyikan isinya.
Plaintext (Teks Terang) adalah suatu pesan yang belum terenkripsi.
Ciphertext (Teks Sandi) adalah suatu pesan yang telah terenkripsi.
Dekripsi adalah suatu proses untuk mengembalikan Teks Sandi menjadi Teks Terang.



Gambar 1. Proses Enkripsi/Dekripsi

2.2 Protokol Kriptografi

Protokol Kriptografi (Schneier. 1996) adalah protokol yang menggunakan mekanisme kriptografi untuk mencapai beberapa fungsi yang terkait keamanan informasi. Protokol kriptografi dibangun dengan melibatkan beberapa algoritma kriptografi.
Protokol kriptografi memiliki karakteristik sebagai berikut:
1) Beberapa pihak yang bertukar pesan;
2) Pihak-pihak ini berusaha untuk mencapai beberapa hal terkait fungsi keamanan;
3) Beroperasi di lingkungan yang tidak bersahabat dan tidak aman;

Protokol harus kuat dan dapat diandalkan dalam menghadapi suatu metode penyerangan yang telah diperkirakan sebelumnya.

2.3 Protokol *Hybrid Cryptosystem*

Protokol kriptografi merupakan protokol yang dibangun dengan melibatkan beberapa algoritma kriptografi (Schneier, 1996). Protokol yang berkembang saat ini telah menggabungkan penggunaan antara sistem penyandian simetrik dan sistem penyandian asimetrik. Penggabungan 2 (dua) sistem tersebut menghasilkan sistem yang disebut dengan *hybrid cryptosystem.*

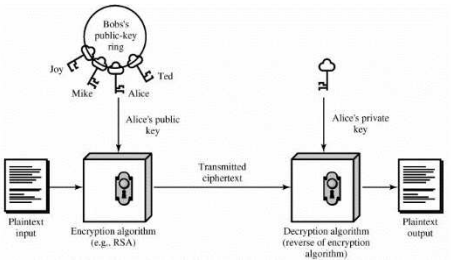
Pembuatan protokol dengan menggunakan *hybrid cryptosystem* dimaksudkan untuk memecahkan permasalahan *key establishment* (mekanisme penggunaan kunci yang disepakati) selain masalah kerahasiaan pesan. Selain itu, masih terdapat permasalahan lain, yaitu masalah keaslian pesan, keaslian entitas dan penyangkalan pesan. Sehingga untuk mengantisipasi semua permasalahan tersebut, banyak protokol yang menggabungkan berbagai sistem kriptografis, yaitu sistem simetrik, sistem asimetrik, fungsi Hash dan sistem pembangkit bilangan acak.

Protokol berdasarkan tujuannya dibedakan menjadi 3 (tiga) macam,yaitu Protokol Otentikasi, Protokol *Key Establishment* dan Protokol *Key Establishment* yang terotentikasi. Pertama, protokol Otentikasi yaitu protokol yang berfungsi untuk menjamin keaslian entitas dari pihak yang lain yang dituju pada saat berkomunikasi. Kedua, Protokol *Key Establishment* yaitu protokol untuk berbagi nilai rahasia bersama. Ketiga, Protokol *Authenticated Key Establishment* yaitu protokol untuk menghasilkan nilai rahasia bersama terhadap pihak yang sudah terjamin keaslian identitasnya (Menezes. 1996).

Entity Authentication adalah proses dimana suatu pihak telah terjamin identitasnya melalui akuisisi bukti yang nyata dari pihak kedua yang terlibat di dalam protokol.

2.4 *Public Key Cryptosystem*

Public Key Cryptosystem adalah suatu sistem kriptografi yang menggunakan dua kunci yang berbeda untuk proses enkripsi-dekripsi yaitu *private key* dan *public key. Public key* bersifat tidak rahasia dan diketahui oleh semua pihak yang bertujuan mengirim pesan kepada penerima. Sedangkan *private key* dirahasiakan oleh penerima (Menezes. 1996).



Gambar 2. Skema *Public Key Cryptosystem*

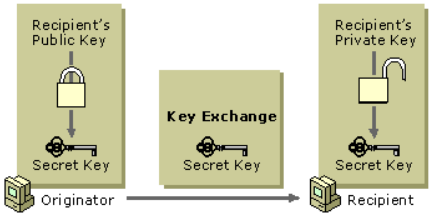
2.5 RSA

Algoritma RSA (Stallings. 2005) merupakan algoritma yang menggunakan *public key cryptosystem.* Dikembangkan pertama kali oleh Rivest, Shamir dan Adleman pada tahun 1977 dengan penggunaan eksponensial. Teks terang dienkripsi pada suatu blok, dimana masing-masing blok memiliki nilai biner kurang dari nilai *n.* Ukuran blok harus bernilai kurang dari atau sama dengan *log(n).* Pada prakteknya ukuran blok adalah *i* bit, dimana $2^i < n < 2^{i+1}$. Enkripsi dan dekripsi ditunjukkan dalam format dibawah ini, untuk beberapa blok teks terang M dan blok teks sandi *C* :

(1) $C = M^e \ mod \ n$

(2) $M = \ C^d \ mod \ n = (M^e)^d \ mod \ n = M^{ed} \ mod \ n$

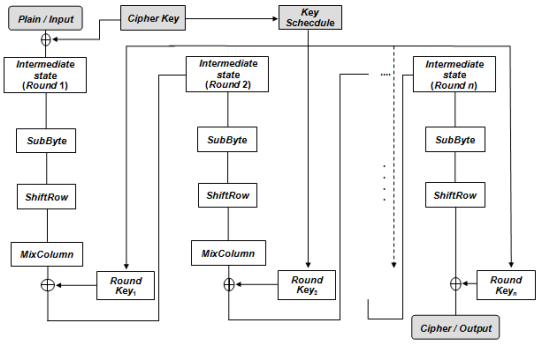
Pada metode RSA *key exchange* (*Secret Key Exchange.* 2014), *secret key* dipertukarkan secara *online* dan aman dengan cara melakukan enkripsi menggunakan *public key* penerima. Hanya penerima yang dituju saja yang dapat melakukan dekripsi terhadap *secret key*, karena proses dekripsi membutuhkan *private key* yang dimiliki oleh penerima. Sehingga pihak ketiga yang melakukan penyadapan terhadap *secret key* yang dienkripsi akan mengalamai kesulitan dalam melakukan dekripsi, seperti yang terlihat pada skema gambar.4 dibawah ini.



Gambar 3. Skema RSA *Key Exchange*

2.6 Algoritma Kriptografi AES

AES (FIPS Publication 197. 2001) merupakan algoritma kriptografi standar yang ditetapkan oleh *National Institute of Standards and Technology* (NIST) melalui publikasi Federal Information Processing Standards (FIPS) 197 pada tahun 2001. AES adalah algoritma kriptografi yang dapat digunakan untuk mengamankan suatu data elektronik. AES merupakan jenis algoritma kriptografi *Block Cipher* dengan kunci kriptografi simetrik. AES dapat menggunakan kunci kriptografi dengan panjang 128-bit, 192-bit atau 256-bit untuk enkripsi atau dekripsi sebuah blok data dengan panjang 128-bit.



Gambar 4. Skema Algoritma AES

2.7 Fungsi Hash

Fungsi *hash* adalah sebuah fungsi yang memetakan suatu string tak terbatas menjadi suatu string dengan panjang tertentu. SHA 256 (*Descriptions of SHA-256, SHA-384, and SHA-512*. 2015) merupakan fungsi hash yang dapat memproses pesan dengan panjang maksimum 2^{64} bit dan menghasilkan output atau *digest* sebesar 256 bit. SHA 256 menggunakan 6 (enam) fungsi logika dimana setiap fungsi tersebut beroperasi pada 32 bit *word* yang direpresentasikan sebagai x, y, dan z. Fungsi logika tersebut merupakan kombinasi dari operasi-operasi dasar seperti AND, OR, XOR, pergeseran bit ke kanan (*shift right*), dan rotasi bit ke kanan (*rotate right*). Fungsi Logika pada SHA 256 yaitu :

$$Ch(x,y,z) = (x \wedge y) \oplus (\neg x \wedge z)$$

$$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$

$$\sum_0^{\{256\}}(x) = ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x)$$

$$\sum_1^{\{256\}}(x) = ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x)$$

$$\sigma_0^{\{256\}}(x) = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x)$$

$$\sigma_1^{\{256\}}(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x)$$

Gambar 5. Fungsi Logika SHA-256

SHA 256 melakukan iterasi fungsi sebanyak 64 kali (*round*).

2.8 *Digital Signature Algorithm* (DSA)

DSA (FIPS Publication 186-3. 2009) merupakan standar algoritma public key NIST yang digunakan untuk melakukan tanda tangan digital. DSA mempunyai 2 (dua) fungsi utama, yaitu Pembangkitan tanda tangan digital (signature generation), dan Verifikasi/pemeriksaan keabsahan tanda tangan digital (*signature verification*). DSA menggunakan 2 (dua) buah kunci yang berbeda dalam proses *signing* dan verifikasi, yaitu public key dan private key. Proses pembangkitan (signing) tanda tangan digital pada algoritma *public key* DSA dilakukan dengan menggunakan private key, sedangkan proses verifikasi (verifying) tanda tangan digital menggunakan *public key*. Proses *signature generation* dan *signature verification* DSA mengacu pada NIST FIPS 186-3.

3. METODE PENELITIAN

Metode Penelitian yang dilakukan pada penelitian ini adalah sebagai berikut :

1) Melakukan studi literatur dan pengumpulan data dari berbagai sumber seperti buku dan internet mengenai penelitian yang akan dilakukan.
2) Melakukan analisis kebutuhan dalam perancangan protocol kriptografi untuk aplikasi *Secure Chat*.
3) Melakukan proses perancangan protokol kriptografi untuk aplikasi *Secure Chat*.
4) Membuat *prototype* aplikasi *Secure Chat* yang dapat berjalan pada *multiplatform* sistem operasi.

4. PEMBAHASAN

4.1 Kebutuhan Protokol Kriptografi

Protokol ini dimaksudkan untuk menjamin keotentikan entitas pengirim dan pesan, serta keamanan informasi yang dikirim pada saat kirim terima pesan (komunikasi) antara user dengan user yang melalui server komunikasi chat. Sehingga berdasarkan hal tersebut, requirement yang dibutuhkan pada protokol, yaitu :

1) Setiap *user* harus membuktikan keaslian identitasnya pada saat sesi awal komunikasi;
2) Pembuktian keaslian identitas pada suatu sesi komunikasi tidak bisa digunakan kembali untuk sesi komunikasi berikutnya;
3) Setiap pesan yang dimodifikasi di tengah jalan dapat langsung teridentifikasi;
4) Kirim-terima pesan hanya bisa dilakukan dengan baik jika menggunakan kunci sesi yang sesuai.
5) Kunci sesi yang telah dibangkitkan hanya dapat digunakan untuk mengamankan komunikasi *chat* pada suatu sesi waktu tertentu.

4.2 Desain Protokol Kriptografi

Protokol kriptografi pada aplikasi *Secure Chat* menitikberatkan kepada masalah jaminan keaslian dan kerahasiaan data yang ditransmisikan. Protokol ini memanfaatkan beberapa algoritma kriptografi yang dikombinasikan dengan menggunakan fungsi hash yang bersifat 1 (satu) arah, untuk tercapainya tujuan keamanan dari *Secure Chat*. Berikut penjelasan mengenai protokol kriptografi pada aplikasi *Secure Chat*:

Kondisi Prasyarat
Misal terdapat 3 (orang) orang pengguna aplikasi *User Secure Chat* yaitu Alice (A), Bob (B) dan Charlie (C), serta seorang pihak ketiga Trent(T) yang dipercaya untuk menjalankan aplikasi Server *Secure Chat*. Kondisi yang ada diasumsikan sebagai berikut:
1) Masing-masing dari A, B, C dan T telah mengetahui kunci public yang digunakan untuk berkomunikasi.
2) Masing-masing dari A, B dan C telah memiliki akun berupa *user*name dan nilai hash password yang tersimpan pada Server milik T.

Penggunaan Notasi
K_{pbA} = kunci public Alice; K_{pkA} = kunci private Alice;
K_{pbB} = kunci public Bob; K_{pkB} = kunci private Bob;
K_{pbT} = kunci public Trent; K_{pkT} = kunci private Trent;
Hash = Fungsi Hash SHA-256;

Langkah-langkah Protokol
Jika Alice ingin berkomunikasi kirim-terima pesan secara aman dengan Bob, maka harus melakukan langkah-langkah sebagai berikut:
1) Langkah-1 → Alice melakukan proses login agar dapat menjalankan aplikasi *User Secure Chat*:
   - Memberi input *User*name dan Password pada form Login;
   - Menghitung nilai Hash(Password);
   - Membangkitkan nilai Random (R);
   - Membentuk *Session Indentifier* (SI) yang dibentuk dari hasil *User*name||Hash(Password)||R;
   - Melakukan proses RSA Sign (tanda tangan) dengan kunci private Alice terhadap SI;
   - Melakukan proses RSA enkripsi terhadap Alice||SI menggunakan kunci public Trent;
   - Mengirimkan hasil enkripsi kepada Trent.

   Hasil yang dikirimkan pada langkah 1 dinotasikan sebagai berikut:
   Alice → Trent : Enkripsi(Alice||Sign(*User*name||Hash(Password)||R)$_{KpkA}$)$_{KpbT}$

2) Langkah-2 → Trent menerima permintaan login dari Alice dan melakukan proses langkah-langkah sebagai berikut :
   - Melakukan proses RSA dekripsi pesan dari Alice menggunakan kunci private Trent menghasilkan Alice||Sign(SI);
   - Melakukan proses RSA Verifikasi terhadap Sign(SI) dengan menggunakan kunci public Alice menghasilkan *User*name||Hash(Password)||R;
   - Melakukan proses otentikasi terhadap *User*name dan Hash(Password), jika terbukti otentik maka Trent akan membangkitkan Timestamp (Ts) dan Identity(ID).
   - Melakukan proses RSA signing terhadap T||ID||R menggunakan kunci private Trent;
   - Melakukan proses RSA enkripsi terhadap Sign(T||ID||R) menggunakan kunci public Alice;
   - Mengirimkan hasil enkripsi kepada Alice.

   Hasil yang dikirimkan pada langkah 2 dinotasikan sebagai berikut:
   Trent → Alice : Enkripsi(Trent||Sign(Ts||ID||R)$_{KpkT}$)$_{KpbA}$

3) Langkah-3 → Alice menerima respon dari Trent dan telah dapat menggunakan aplikasi *Secure Chat*, selanjutnya Alice mengirimkan permintaan untuk dapat berkomunikasi dengan Bob, dan melakukan proses langkah-langkah sebagai berikut :
   - Melakukan proses RSA dekripsi pesan dari Trent menggunakan kunci private Alice menghasilkan → Trent, Sign(Ts||ID||R);

- Melakukan proses RSA Verifikasi terhadap Sign(Ts‖ID‖R) dengan menggunakan kunci public Trent menghasilkan Ts‖ID‖R;
- Mengirimkan hasil enkripsi berupa Alice‖Bob‖Ts‖ID yang merupakan permintaan komunikasi antara Alice-Bob kepada Trent.

Hasil yang dikirimkan pada langkah 3 dinotasikan sebagai berikut:
Alice → Trent : Enkripsi(Alice||Bob||TS||ID)$_{KpbT}$

4) Langkah-4 Bob telah melakukan langkah-1 sampai dengan langkah-2 seperti yang telah dilakukan Alice, dengan melakukan proses login agar dapat menjalankan aplikasi *User Secure Chat*, dan mengirimkan permintaan untuk dapat berkomunikasi dengan Alice.

Hasil yang dilakukan pada langkah 4 dinotasikan sebagai berikut:
Bob → Trent : Enkripsi(Bob||Alice||Ts||ID)$_{KpbT}$

Dari hasil langkah-1 sampai dengan langkah-4 Alice dan Bob telah dapat saling berkomunikasi melalui perantara Trent, akan tetapi kirim-terima masih dilakukan dengan pesan biasa (*plaintext*). Agar Alice dan Bob dapat berkomunikasi secara aman maka diperlukan suatu proses key establishment untuk menetapkan kunci sesi yang akan digunakan untuk proses enkripsi/dekripsi

5) Langkah-5 → Alice mengirimkan permintaan untuk dapat berkomunikasi *secure* dengan Bob, dan melakukan proses langkah-langkah sebagai berikut :
   - Alice membangkitkan Kunci Sesi (K_s) yang akan digunakan dalam komunikasi *secure* dengan Bob;
   - Melakukan proses RSA enkripsi terhadap Sign(T‖ID‖ K_s) menggunakan kunci public Bob;
   - Mengirimkan hasil enkripsi berupa permintaan komunikasi antara Alice-Bob kepada Trent;
   - Trent meneruskan pesan enkripsi kepada Bob.

Hasil yang dikirimkan pada langkah 5 dinotasikan sebagai berikut:
Alice → Trent : Enkripsi(Alice‖Bob‖TS‖ID)$_{KpbT}$, Enkripsi(Alice‖Bob‖Sign(Ts‖ID‖K_s)$_{KpkA}$)$_{KpbB}$
Trent → Bob : Enkripsi(Alice||Bob||Sign(Ts||ID||K_s)$_{KpkA}$)$_{KpbB}$

6) Langkah-6 Bob melakukan proses RSA dekripsi dari pesan *Enkripsi(Alice||Bob||Sign(Ts||ID||K_s)$_{KpkA}$)$_{KpbB}$* yang diterima dari Trent untuk mendapatkan K_s yang akan digunakan untuk proses enkripsi/dekripsi pada saat kirim-terima pesan melalui *Secure Chat* antara Alice dengan Bob. Konfirmasi penerimaan Ks oleh Bob dilakukan dengan mengirimkan hasil enkripsi Ts‖ID menggunakan kunci Ks kepada Alice.

Hasil yang dilakukan pada langkah 4 dinotasikan sebagai berikut:
Bob : ~~Dekripsi~~(~~Enkripsi~~(Alice‖Bob‖Sign(Ts‖ID‖K_s)$_{KpkA}$)$_{~~KpbB~~}$) $_{~~KpkB~~}$
: ~~Verifikasi~~(~~Sign~~(Ts‖ID‖K_s)$_{~~KpkA~~}$) $_{~~KpbA~~}$
: Ts‖ID‖K_s

Bob → Alice : Enkripsi(Ts||ID)K_s
: ~~Verifikasi~~(~~Sign~~(Ts‖ID‖K_s)$_{~~KpkA~~}$) $_{~~KpbA~~}$
: Ts‖ID‖K_s

7) Langkah-7 → Alice dan Bob telah memiliki kunci K_s yang sama, sehingga komunikasi *Secure Chat* antara Alice dan Bob dapat dilakukan dengan proses AES enkripsi/dekripsi menggunakan kunci K_s, serta tambahan nilai hash dari Message (pesan), untuk memeriksa keutuhan data

Alice → Trent : Enkripsi(Message1)$_{Ks}$‖Hash(Message1);
Trent → Bob : Enkripsi(Message1)$_{Ks}$‖Hash(Message1);
Bob : ~~Dekripsi~~(~~Enkripsi~~(Message1)$_{~~Ks~~}$) $_{~~Ks~~}$||Hash(Message1) → Message1‖Hash$_1$(Message1)

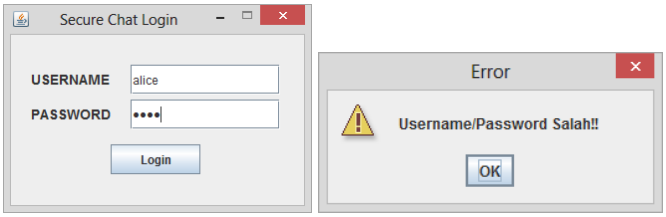
Bob → Trent : Enkripsi(Message2)$_{Ks}$‖Hash(Message2);
Trent → Alice : Enkripsi(Message2)$_{Ks}$‖Hash(Message2);

Alice : ~~Dekripsi~~(~~Enkripsi~~(Message2)$_{~~Ks~~}$) $_{~~Ks~~}$||Hash(Message2) → Message2||Hash(Message2)

4.3 Implementasi *Secure Chat*

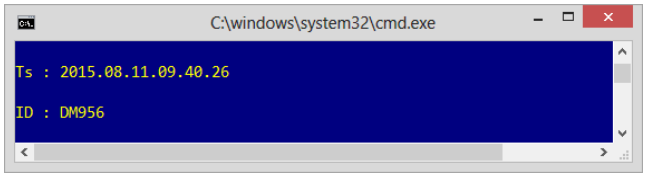
Pada tahap ini yang dilakukan adalah melakukan implementasi dari desain protokol kriptografi untuk mengamankan proses kirim terima pesan antara user aplikasi Secure Chat, proses autentikasi user dan key establishment berjalan pada background aplikasi yang akhirnya bertujuan untuk mengamankan komunikasi kirim-terima pesan antara user melalui server. Implementasi dibuat dalam bentuk simulasi menggunakan Bahasa Pemograman Java Enterprise Edition 8 SDK serta sistem operasi Windows 8 dan Linux Ubuntu 12. Pemilihan bahasa pemrograman Java dikarenakan sasaran penggunaan protokol program agar dapat digunakan pada berbagai sistem operasi. Simulasi implementasi akan dibuat berurut sesuai langkah pada tahap desain.

1) Langkah-1 → Alice melakukan proses login agar dapat menjalankan aplikasi *User Secure Chat*:



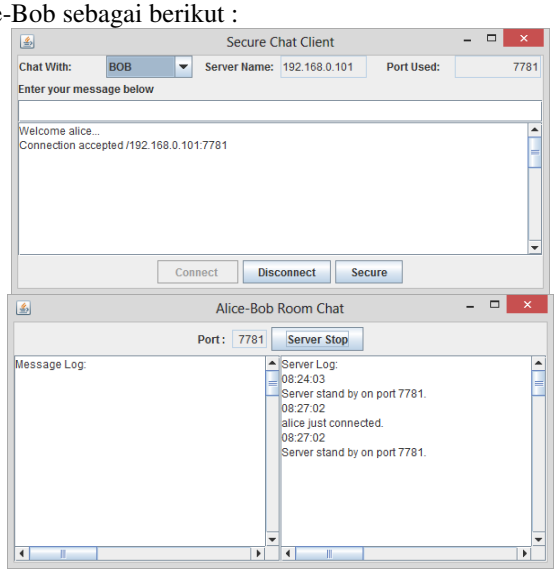
Gambar 6. Tampilan Form Login dan Error Jika Login Gagal

2) Langkah-2 → Trent menerima permintaan login dari Alice dan memberikan pengenal waktu berupa Timestamp (Ts) dan pengenal identitas berupa random ID sebagai berikut :



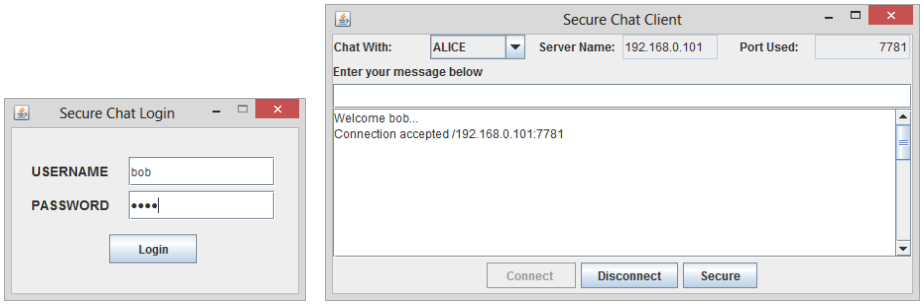
Gambar 7. Hasil Pembangkitan *Timestamp* dan ID

3) Langkah-3 → Alice menerima respon dari Trent dan telah dapat menggunakan aplikasi *Secure Chat*, selanjutnya Alice mengirimkan permintaan untuk dapat berkomunikasi dengan Bob, dan Trent merespon dengan mengalokasikan suatu *chat* room dengan port tertentu yang dapat digunakan untuk komunikasi antara Alice-Bob sebagai berikut :



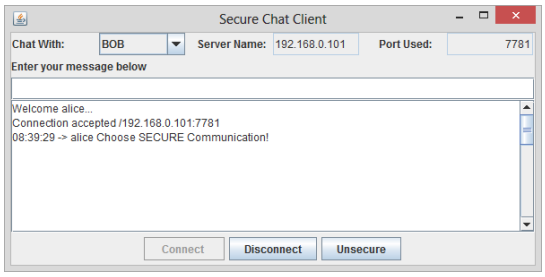
Gambar 8. Tampilan Secure Chat Client dan Server Chat Room

4) Langkah-4 Bob telah melakukan langkah-1 sampai dengan langkah-2 seperti yang telah dilakukan Alice, dengan melakukan proses login agar dapat menjalankan aplikasi *User Secure Chat*, dan mengirimkan permintaan untuk dapat berkomunikasi dengan Alice.



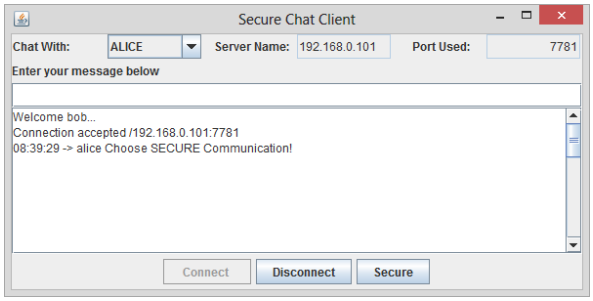
Gambar 9. Tampilan Form Login dan Secure Chat Client Bob

5) Langkah-5 → Alice mengirimkan permintaan untuk dapat berkomunikasi *secure* dengan Bob sebagai berikut :



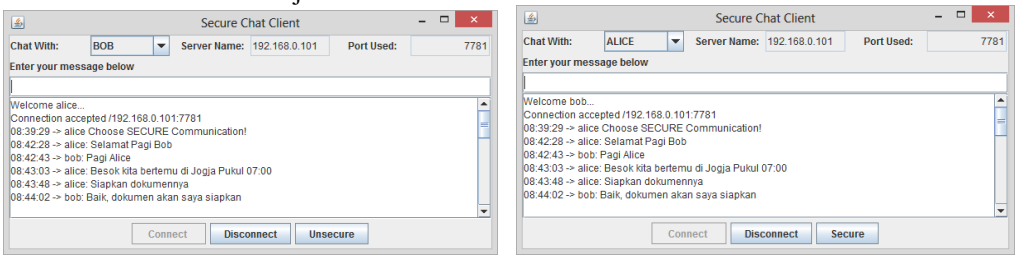
Gambar 10. Permintaan Secure Communication dari Alice

6) Langkah-6 Bob melakukan proses dekripsi pesan untuk mendapatkan K_s yang akan digunakan untuk proses enkripsi/dekripsi pada saat kirim-terima pesan melalui *Secure Chat* antara Alice dengan Bob.



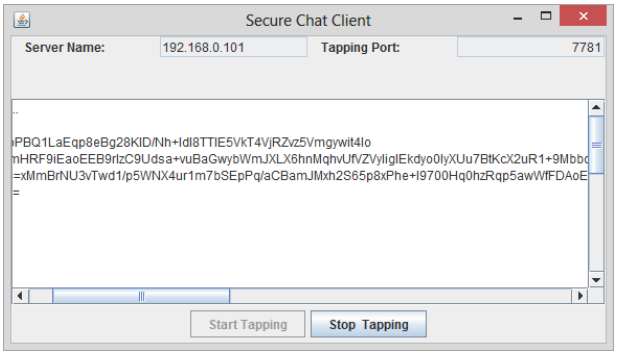
Gambar 11. Bob Menerima Permintaan Secure Communication dari Alice

7) Langkah-7 → Alice dan Bob telah memiliki kunci K_s yang sama, sehingga komunikasi *Secure Chat* antara Alice dan Bob telah berjalan



Gambar 12. Tampilan *Secure Communication* antara Alice dan Bob

Komunikasi *Secure Chat* antara Alice dengan Bob jika dilakukan *tapping* (penyadapan) hasilnya merupakan karakter yang tidak dapat terbaca seperti yang terlihat pada gambar 13. di bawah ini.



Gambar 13. Tampilan *Secure Communication* antara Alice dan Bob jika tersadap

Tampilan implementasi sebelumnya merupakan implementasi pada sistem operasi Windows 8. Pada gambar di bawah ini merupakan tampilan aplikasi *Secure Chat* pada sistem operasi Linux Ubuntu 12.

Gambar 14. Tampilan Implementasi Pada Linux Ubuntu 12

5. KESIMPULAN

1) Protokol Kriptografi adalah protokol yang menggunakan mekanisme kriptografi untuk mencapai beberapa fungsi yang terkait keamanan informasi. Protokol kriptografi dibangun dengan melibatkan beberapa algoritma kriptografi.
2) Protokol Kriptografi pada aplikasi *Secure Chat* menjamin keotentikan entitas pengirim dan pesan, serta keamanan infromasi yang dikirim pada saat kirim terima pesan (komunikasi) antara *client* dengan *server*
3) Implementasi dari aplikasi *Secure Chat* dibuat menggunakan bahasa pemrograman Java agar dapat berjalan pada *multiplatform* sistem operasi

DAFTAR PUSTAKA

FIPS Publication 186-3 (2009). *Digital Signature Standard* (DSS). National Institute of Standards and Technology.

FIPS Publication 197 (2001). *Announcing The Advanced Encryption Standard* (AES). National Institute of Standards and Technology.

Menezes, J. Alfred et al. 1996, *Handbook ofApplied Cryptography*. CRC Press.

Schneier, Bruce. 1996, *Applied Cryptography, Second Edition*. John Wliey & Sons Inc

Stallings, William. 2005, *Cryptography and Network Security Principles and Practices, Fourth Edition*, *Prentice Hall*.

Descriptions of SHA-256, SHA-384, and SHA-512, http://www.iwar.org.uk/ . Diakses tanggal 24 Juli 2015

Secret Key Exchange, *http://technet.microsoft.com/en-us/library/cc962035.aspx* . Diakses tanggal 12 April 2014.