Optimasi Keamanan *Instant message* pada Sistem Operasi Android

Mukhlison, Sholeh Hadi Pramono, dan Onny Setyawati

Abstract – **In this study AES and Kerberos is used to optimize the security system on instant message so that a text message is sent securely through on goal. The addition and incorporation of AES with Kerberos is used to improve the efficiency of security primarily on user authentication and file encryption text message. The test results showed a AES and kerberos able to optimize the security system instant message. Percentage of instant message security system applications in the study reached 95 %, derived from the percentage of test data and the success rate error test results. AES - 256 had 14 round keys with a key length of 256 - bits, hence AES encrypted file has a powerful message.**

Index Terms—AES, Kerberos, security system, instant message.

***Abstrak*—Optimasi keamanan *instant message* menggunakan AES (*Advanced encryption standard*) dan Kerberos merupakan solusi untuk mengatasi kelemahan dari sistem keamanan *instant message* yang mengandalkan sistem otentikasi *password based* dan sitem enkripsi yang lemah. Sistem keamanan *instant message* yang sudah berjalan masih dinilai rawan terhadap beberapa serangan seperti *brute-force* dan *sniffing*, sedangkan AES dan kerberos memberikan jaminan otentikasi yang handal dan sistem enkripsi yang sangat kuat sehingga tahan terhadap berbagai serangan cyber seperti *brute-force* dan *sniffing*. Pada penelitian ini AES dan kerberos digunakan untuk mengoptimalkan sistem keamanan pada *instant message* sehingga pesan teks terkirim dengan aman sampai pada tujuan. Penambahan dan penggabungan AES dengan Kerberos digunakan untuk meningkatkan efisiensi keamanan terutama pada otentikasi pengguna dan enkripsi file pesan teks yang dikirim. Hasil pengujian menunjukkan AES dan kerberos mampu mengoptimalkan sitem keamanan pada *instant message*. Persentase keamanan sistem aplikasi *instant message* dalam penelitian ini mencapai 95%. yang diperoleh dari persentase data uji tingkat keberhasilan dan error hasil pengujian. Kerberos tidak mengirimkan password ke server dalam sistem otentikasinya sehingga password tidak mudah dicuri. AES-256 memiliki 14 putaran kunci dengan panjang kunci 256-bit, sehingga AES memiliki hasil enkripsi file pesan yang kuat. Dengan panjang kunci sebesar 256-bit, maka terdapat sebanyak 2^{256} = 1,1 x 10^{77} kemungkinan kunci.**

Mukhlison adalah Mahasiswa Program Pascasarjana Teknik Elektro Universitas Brawijaya, Malang, Indonesia (email mukhlisonst@gmail.com).

Sholeh Hadi Pramono adalah Dosen Jurusan Teknik Elektro Universitas Brawijaya Malang.

Onny Setyawati adalah Dosen Teknik Elektro Universitas Brawijaya Malang

***Kata Kunci*— AES, Kerberos, Sistem *K*eamanan, *Instant message*.**

I. PENDAHULUAN

INSTANT MESSAGE adalah suatu sistem pengiriman pesan dengan cepat melalui perantara jaringan internet dari satu komputer ke komputer yang lain. *Instant message* merupakan sebuah teknologi internet yang memungkinkan para pengguna dalam jaringan internet untuk mengirimkan pesan singkat secara langsung pada saat yang bersamaan (*real time*) dengan menggunakan teks kepada pengguna lainnya yang sedang terhubung ke jaringan yang sama. Karena sifat dari *instant messaging* tersebut yang langsung (*real time*) dan dua arah, sebagian besar pengguna menganggap bahwa *instant messaging* dapat meningkatkan produktivitas dalam pekerjaan [1]. Ancaman serangan penyadapan yang akhir- akhir ini sering terjadi pada jaringan *instant message* merupakan sesuatu hal yang mengkhawatirkan bagi pengiriman data rahasia yang melalui jaringan.

Banyak penelitian yang telah dikembangkan untuk memperbaiki kelemahan keamanan pada *instant message* diantaranya menggunakan fitur *Hidden chat* yang ada pada Line dan penggunaan enkripsi md5 hash yang ada whatsapp pada android. Sistem keamanan menggunakan hidden chat dan algoritma enkripsi md5 hash masih dianggap lemah karena masih sering terjadi penyadapan dan fabrikasi terhadap pesan teks yang terkirim oleh orang yang tidak bertanggungjawab.

Dengan demikian, maka keamanan menjadi suatu hal yang sangat penting untuk dipertimbangkan. Apalagi jika data yang dikirimkan via layanan *instant messaging* tersebut sifatnya rahasia. Maka dari itu peran utama kriptografi untuk mengamankan data dari berbagai ancaman penyadapan dengan menggunakan teknik enkripsi sangatlah penting.

Salah satu algoritma enkripsi yang sering dipakai dalam pengenkripsian file yaitu AES (*Advanced encryption standard*). AES merupakan algoritma kriptografi simetri, ini berarti proses enkripsi dan dekripsinya harus menggunakan kunci yang sama. Hal tersebut tentunya bukan hal yang dapat dianggap remeh, masalah ini dapat di selesaikan dengan menggunakan sistem otentikasi pengguna yang tepat sehingga dapat meningkatkan keamanan pesan teks yang dikirim, salah

satu sistem otentikasi yang cukup handal adalah Kerberos.

Protokol Kerberos adalah protokol otentikasi jaringan yang dikembangkan oleh MIT (*Massachusetts Institute Technology)*. Protokol Kerberos ini menggunakan kriptografi untuk sistem otentikasinya, baik sisi *client* maupun server, sehingga protocol Kerberos mampu mengatasi berbagai kelemahan dari sistem otentikasi konvensional (*password-based)* [2]. Kerberos menyediakan sistem otentikasi data pengguna yang handal untuk kedua belah host yang saling berkomunikasi. Otentikasi data adalah jaminan bahwa komunukasi kedua pihak berjalan dengan baik dan benar [3].

Untuk meningkatkan keamanan pengiriman pesan teks dalam *instant messaging* agar tahan terhadap berbagai serangan yang mungkin akan mengancam, maka penulis membuat sebuah sistem keamanan yang dapat mengoptimasi keamanan pengiriman pesan teks dengan mengimplementasikan Kerberos dan AES pada proses pengiriman pesan teks.

Fokus dalam penelitian ini adalah optimasi kemanan *instant message* menggunakan protokol otentikasi yang handal dan algoritma enkripsi yang terstandarisasi dan handal. Metode yang diusulkan adalah protocol otentikasi Kerberos dan algoritma enkripsi simetri yang handal yaitu AES. Kerberos digunakan untuk meningkatkan sistem otentikasi pengguna sehingga bisa mengoptimalkan sistem otentikasi password-based dan mendapatkan solusi optimum global. AES pada *instant message* berfungsi untuk mengenkripsi pesan teks yang dapat mencegah fabrikasi pesan teks.

II. KRIPTOGRAFI

Kriptografi (*cryptography*) berasal dari bahasa Yunani yang terdiri dari kata *cryptos* yang artinya *secret* (tersembunyi atau rahasia) dan *graphia* yang artinya *writing* (sesuatu yang tertulis) sehingga kriptografi dapat juga disebut sebagai sesuatu yang tertulis secara rahasia atau tersembunyi. *The Concise Oxford Dictionary* mendefinisikan kriptografi sebagai seni menulis atau memecahkan kode.

Dalam kriptografi, pesan atau informasi yang dapat di baca disebut sebagai *plaintext* atau *clear text*. Proses yang dilakukan untuk mengubah *plaintext* ke dalam *chipertext* disebut enkripsi. Pesan yang tidak terbaca disebut *ciphertext*. Proses kebalikan dari enkripsi disebut dekripsi. Dekripsi akan mengembalikan *ciphertext* menjadi *plaintext*. Kedua proses enkripsi dan dekripsi membutuhkan penggunaan sejumlah informasi rahasia, yang sering disebut kunci (*key*). Jadi terdapat tiga fungsi dasar dari kriptografi yaitu : enkripsi, dekripsi dan kunci [4].

A. Kerberos

Kerberos adalah protokol otentikasi jaringan yang dikembangkan oleh MIT. Kerberos adalah protokol otentikasi yang menggunakan pihak ketiga yang dipercaya bersama-sama (*trusted third-party*). Protokol ini menggunakan kriptografi untuk otentikasi baik sisi *client* maupun server, sehingga protokol ini mampu mengatasi kelemahan dari sistem otentikasi *password-based.*

Perancangan Kerberos ditujukan untuk memberikan solusi bagi serangan-serangan keamanan yang tidak dapat diatasi oleh sistem otentikasi konvensional. Beberapa serangan yang ingin diatasi dengan perancangan Kerberos, antara lain:

1. *impersonation*, yaitu menggunakan *username* dan *password* yang bukan miliknya untuk memperoleh akses *service* dari jaringan.
2. *eavesdroping*, yaitu menyadap data-data yang lalu lalang di jaringan (*passive attack*).
3. *tampering*, yaitu mengambil data-data yang lalu lalang, mengubahnya,lalu mengirimkannya kembali (integritas data berubah).

Kerberos dalam keamanan komputer adalah merujuk kepada sebuah protocol otentikasi yang dikembangkan oleh *Massachusetts Institute Technology* (MIT). Protokol Kerberos memiliki tiga subprotokol agar dapat melakukan aksinya:

1. *Authentication Service (AS) Exchange*
2. *Ticket-Granting Service (TGS) Exchange*
3. *Client/*Server*(CS)Exchange*

Cara kerja Kerberos melakukan otentikasi dapat dibagi menjadi empat tahap.

- Tahap pertama disebut *Authentication Exchange*
- Tahap berikutnya disebut *TGS Exchange*
- Tahap ketiga disebut *Client/*Server *Exchange*
- Tahap terakhir disebut *Secure Communication*

B. AES *(Advanced Encryption Standard)*

AES (*Advanced Encryption Standard*) adalah *cipher* blok yang menggantikan DES. Algoritma AES menggunakan substitusi dan permutasi, dan sejumlah putaran (*cipher* berulang), dimana setiap putaran menggunakan kunci yang berbeda (kunci setiap putaran disebut *round key*).*Input* dan *output* dari algoritma AES terdiri dari urutan data sebesar 128 bit. Urutan data yang sudah terbentuk dalam satu kelompok 128 bit tersebut disebut juga sebagai blok data atau *plaintext* yang nantinya akan dienkripsi menjadi *ciphertext* [5].

a. Proses Enkripsi

1. *AddRoundKey:* melakukan XOR antara state awal (plainteks) dengan cipher key. Tahap ini disebut juga initial round.
2. Putaran sebanyak Nr – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   a. *SubBytes:* substitusi byte dengan menggunakan tabel substitusi (S-box).
   b. *ShiftRows:* pergeseran baris-baris array state secara wrapping.
   c. *MixColumns: mengacak data di masing-masing* kolom *array state.*
   d. *AddRoundKey: melakukan XOR antara state* sekarang *round key.*

B. Proses Dekripsi

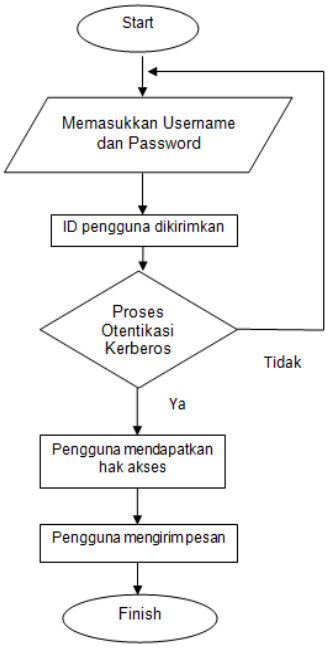
1. *InvShiftRows* : transformasi byte yang berkebalikan dengan transformasi *ShiftRows*.
2. *InvSubBytes :* merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*.
3. *InvMixColumns :* Setiap kolom dalam *state* dikalikan dengan matrik perkalian dalam AES.

III. METODE PENELITIAN

Dalam penelitian ini untuk memperoleh hasil penelitian menurut cara memperolehnya akan menggunakan data primer, dimana data yang nantinya akan digunakan adalah data-data performa yang dihasilkan dari eksperimen menggunakan AES dan kerberos.

Sistem enkripsi *instant message* ini menitik beratkan kepada kemampuan sebuah sistem untuk mengotentikasi pengguna dan mengenkripsi data pesan teks yang dikirim melalui jaringan.

Dari Gambar.1 dapat dipahami bahwa, proses awal otentikasi Kerberos diawali dengan memasukkan *username* dan *password* pada form *login*. Hal ini sebagai syarat masuk pengguna ke dalam sistem aplikasi. Setelah *username* dan *password* diterima oleh server kerberos, maka proses otentikasi dimulai. Apabila *username* dan *password* dianggap valid atau terdaftar oleh server kerberos, maka pengguna dapat meneruskan proses selanjutnya yaitu masuk ke dalam sistem aplikasi.



Gambar .1. *Flowchart* Otentikasi Kerberos

Cara kerja Kerberos melakukan otentikasi dapat dibagi menjadi empat tahap. Tahap pertama disebut *Authentication Exchange*. Pihak yang terlibat adalah *client* dan Kerberos *Authentication* Server (AS). Untuk *login* kejaringan, program di sisi *client* (dikenal dengan kinit) akan meminta *user* untuk memasukkan *username* dan *password*. Program ini akan menurunkan *client key* (KC) dari *password* dan menghapus *password* sebenarnya di*workstation* tersebut. *Username* akan dikirim melintasi jaringan ke AS. Jika *username* terdapat di *database*, maka AS akan membuat *Session Key* 1 (SK1atau KC,TGS) untuk komunikasi antara *client* dan *Ticket-granting* Server (TGS).

Selain itu, AS juga membuat *ticket* untuk komunikasi antara *client* dan TGS (disebut *Ticket-granting Ticket* atau TGT atau TC, TGS). Selanjutnya KC ,TGS dan TC, TGS dienkripsi dengan *TGS key* (KTGS). Paket ini diperuntukkan untuk dibuka hanya oleh TGS. Paket TGS dan KC, TGS dienkripsi dengan KC, lalu dikirimkan ke *client*. Notasi untuk proses ini dapat ditunjukkan seperti dibawah ini:

{ KC,TGS, { KC,TGS, TC,TGS }KTGS} KC

dimana { TX }KX berarti *ticket* TX dienkripsi dengan kunci KX. Yang bergaris bawah menunjukkan paket TGS yang dienkripsi dengan KTGS.

Tahap berikutnya disebut *TGS Exchange. Data* dari AS didekripsi dengan menggunakan KC. Jika *password* yang dimasukkan sesuai dengan *username*, maka *client* akan mampu mendekripsi data dengan benar. *Client* akan mendapatkan KC,TGS dan paket TGS yang masih dalam keadaan terenkripsi. *Client* tidak dapat membuka paket ini karena kunci yang dipakaidalah KTGS, yang hanya diketahui oleh AS dan TGS. Selanjutnya *client* akan membuat *Authenticator* (Auth atau AC) yang berisi *username*, IP *address client*, dan *time-stamp*. Lalu *client* akan mengirimkan nama server yang dituju(S), AC, dan mem-*forward* paket TGS dari AS ke TGS melintasi jaringan.

Notasi dari pengiriman tersebut adalah sebagai berikut:

S, { AC }KC,TGS, { KC,TGS, TC,TGS }KTGS

Di TGS paket TGS dari AS didekrip dengan KTGS dan TGS memperoleh KC, TGS dan TTGS. KC,TGS digunakan untuk mendekrip AC. Jika isi AC dan TC,TGS sesuai, maka TGS akan memberi akses dengan cara membuat *Session Key*2 (SK2 atau KC,S) untuk komunikasi antara *client* dan server yang dituju(disebut juga *Target* Server atau TS). TGS akan mengeluarkan *ticket* baru (disebut TK-TS atau TC,S). TC,S dan KC,S akan dienkrip dengan kunci privat server (KS) menjadi paket TS dari TGS. KC,S dan paket TS dienkrip dengan KC, TGS, kemudian dikirimkan ke *client* melintasi jaringan. Notasi untuk pengiriman ini dinyatakan sebagai berikut:

{ KC,S, { KC,S, TC,S }KS}KC, TGS bergaris bawah menunjukkan paket TS dari TGS.

Tahap ketiga disebut *Client/*Server *Exchange*. Pada tahap ini *client* dan server yang bersangkutan akan melakukan otentikasi. Otentikasi ini dapat berlangsung searah atau dua arah (*mutual authentication*). Otentikasi searah berarti *client* harus membuktikan ke server siapa dirinya, sedangkan pada otentikasi dua arah server juga harus membuktikan kepada *client* siapa dirinya.*Client* mendekrip data yang diterima dengan KC,TGS dan mendapatkan KC,S dan paket TS dari TGS. Paket TS ini tidak dapat dibuka oleh *client* karena proses dekripsi dilakukan dengan menggunakan kunci privat KS yang hanya diketahui oleh TGS dan TS. Kemudian *client*

akan membuat AC, dan mengenkripsinya dengan KC,S. Selanjutnya *client* mengirimkan AC tersebut dan mem-*forward* paket TS dari TGS ke server *instant message* yang dituju melintasi jaringan.
Notasi untuk proses ini dinyatakan sebagai berikut:

{ AC }KC,S, { KC,S, TC,S }KS

Sesampainya di server *instant message*, server akan mendekrip paket TS dari TGS dengan kunci privat yang dimilikinya, dan mendapatkan KC,S dan TC,S. KC,S digunakan untuk mendekrip AC. Jika isi AC dan TC,S sesuai, maka TS akan memberi akses kepada *client* untuk mendapatkan *service* darinya. Dengan demikian TS telah diyakinkan bahwa *user* yang meminta *service* padanya adalah *user* yang sah. Jika *mutual authentication* dibutuhkan, maka TS akan mengirimkan data *time stamp* yang tercantum di AC ditambah satu, lalu dienkripsi dengan *session key* KC,S. Notasinya adalah sebagai berikut:

{ time-stamp AC+ 1 }KC,S

Dengan demikian kedua belah pihak diyakinkan akan kebenaran identitas masing-masing.

Tahap terakhir disebut *Secure Communication*. Baik *client* dan TS (Target server/Server *instant message*) telah diyakinkan akan kebenaran identitas masing-masing. Pertukaran data diantara keduanya dapat dilakukan dengan aman karena *client* dan server *instant message* memiliki kunci privat KC,S yang hanya diketahui oleh mereka saja.

A. Algoritma AES

Algoritma AES memiliki keunggulan waktu dalam proses enkripsi dan dekripsinya dibandingan dengan algoritma lain maka dari itu algoritma AES ini sangat tepat digunakan dalam *instant messaging* yang bersifat langsung (*real time*).

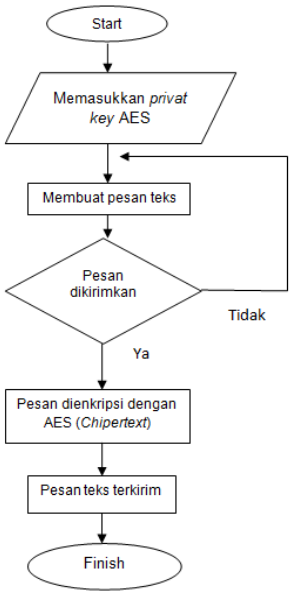
AES mampu untuk mendeteksi, jika pesan telah rusak atau dirusak selama transmisi. Dan AES dapat mengamankan pesan yang berisi informasi rahasia /sensitif yang disimpan untuk tetap dirahasiakan, bahkan ketika perangkat diakses oleh orang yang tidak bertanggungjawab. Data yang terenkripsi dengan AES tahan terhadap berbagai serangan seperti serangan *brute force* dan *pattern attack* (Rohan dkk,2012).

Langkah pertama dalam proses enkripsi AES adalah pemrosesan plainteks menjadi chiperteks. Ada beberapa proses dalam pengenkripsian pesan teks ,yaitu:

a. Enkripsi Pesan Teks

Proses awal enkripsi teks adalah dengan memasukkan *plaintext* yang akan dikirimkan ke penerima pesan yang disertakan dengan kunci *Random*. Proses selanjutnya nantinya akan sama dengan penjelasan proses enkripsi algoritma AES seperti flowchart pada gambar 2 berikut.

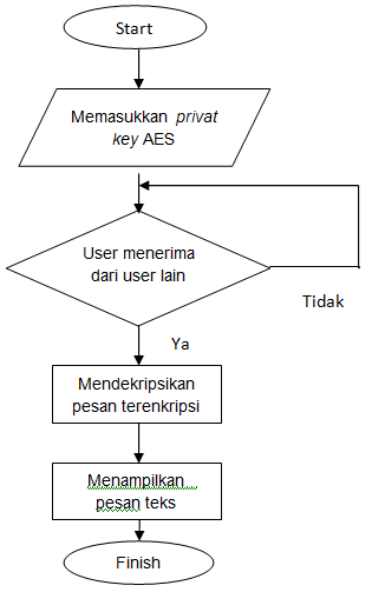
Gambar 2 berisi proses pengenkripsian pesan teks pengiriman pesan ini, dimulai dengan memasukkan key atau kunci sebagai *private key* dari algoritma enkripsi AES. Kunci ini jugalah yang nantinya akan digunakan untuk mendekripsikan file pesan teks terkirim. Proses selanjutnya yang dilakukan oleh *user*/pengguna adalah membuat pesan teks. Pesan teks bisa berupa kata, kalimat ataupun paragraf sesuai dengan kode ASCII. Jumlah karakter maksimal isi pesan adalah 256 word dalam sekali pengiriman. Selanjutnya pesan akan melewati proses pengiriman pesan, pesan yang terkirim sebelumnya akan dienkripsi terlebih dahulu dengan algoritma enkripsi AES. Setelah proses enkripsi pesan teks dengan algoritma AES selesai, maka pesan teks akan langsung dikirim ke penerima.



Gambar .2. *Flowchart* Enkripsi Pesan Teks

b. Dekripsi Pesan Teks

Proses awal dekripsi teks adalah dengan memasukkan *ciphertext* yang telah dienkripsi pada proses enkripsi diawal tadi yang disertakan dengan kunci *Random*. Proses selanjutnya nantinya akan sama dengan penjelasan proses dekripsi algoritma AES seperti flowchart pada Gambar 3 berikut.



Gambar. 3. *Flowchart* Dekripsi Pesan Teks

Dari Gambar .3. dapat dipahami bahwa, pada proses dekripsi pesan teks pada sistem aplikasi keamanan *instant message* ini, proses akan diawali dengan memasukkan *privat key*. *Private key* ini akan digunakan untuk membuka atau mendekripsikan pesan yang telah diterima dari pengirim pesan yang berupa pesan terenkripsi.

Proses selanjutnya adalah pendekripsian yang dilakukan oleh sistem dengan menggunakan *private key* AES yang telah dimasukkan. Selanjutnya pesan yang telah didekripsikan tersebut akan ditampilkan, sehingga penerima pesan dapat membaca isi pesan teks yang dikirimkan. Pesan yang didekripsi harus sama hasilnya dengan pesan yang dikirimkan oleh pengirim.

B. *Desain*

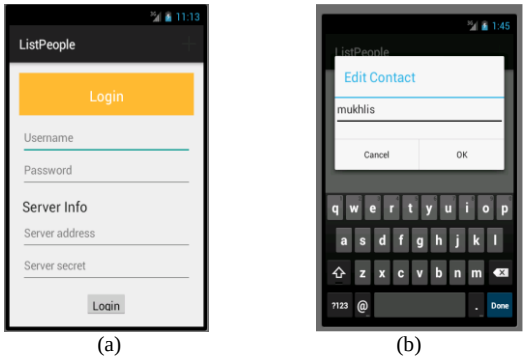
Desain sistem aplikasi ini dimulai dengan memasukkan *username* dan *password*, jumlah karakter maksimal *username* dan *password* adalah 30 karakter. *Username* dan *password* yang telah dimasukkan selanjutnya akan diotentikasi oleh Kerberos, apabila hasil otentikasi menyatakan bahwa *username* dan *password* valid maka pengguna diizinkan untuk masuk pada sistem dan apabila hasilnya invalid maka proses akan diulang untuk memasukkan *username* dan *password* lagi.

IV. HASIL DAN PEMBAHASAN

A. *Hasil Pengujian Aplikasi*

Gambar .4 menunjukkan program aplikasi instant message, halaman *login* merupakan halaman pertama *user* untuk mulai menggunakan aplikasi *instant message*. Pengguna harus memasukkan *username*, *password*, server *address* dan server secret (*private key* AES). Halaman selanjutnya adalah halaman *addfriends*, yang digunakan untuk menambahkan teman dalam komunikasi pesan yang diinginkan.

Halaman *Listfrends*, menunjukkan teman yang ada pada daftar teman siap untuk komunikasi. halaman ini adalah halaman daftar teman yang telah ditambahkan di halaman *add friend*. Halaman pesan, menunjukan halaman yang digunakan untuk komunikasi transmisi data teks kepada *user* lain yang terhubung pada jaringan server. Hasil pengujian aplikasi menunjukkan semua bagian sistem berjalan sempurna tanpa error.

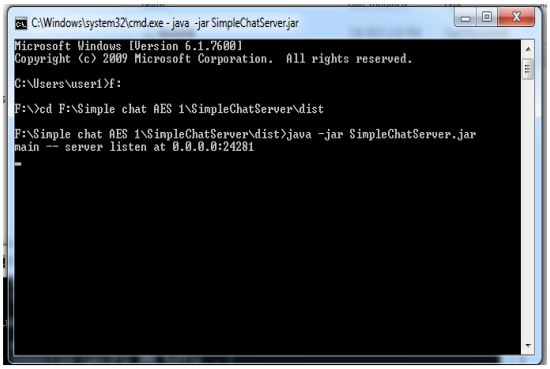


(a) (b)

Gambar .4. Tampilan Aplikasi (a. Halaman Login, b. Halaman *add friend*)

B. *Hasil Pengujian* Server

Pengujian Berikut ini adalah pengujian server *Instant message* dan server Kerberos. Gambar.5 ini menunjukkan bahwa server *instant message* dan Kerberos siap untuk digunakan pada proses pengiriman pesan melalui *instant message*, ditunjukkan dengan command "*main* – server *listen at* 0.0.0.0:24281".

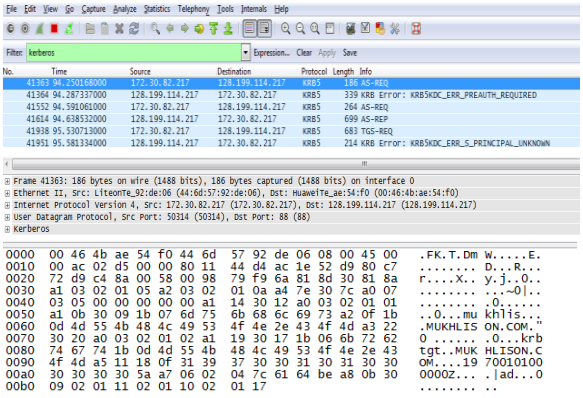


Gambar .5. Tampilan Server I*nstant Message*

C. *Hasil Analisa Pengujian Kerberos*

Proses otentikasi jaringan dengan menggunakan Kerberos terpusat pada server Kerberos. Setiap proses yang ada di *instant message* akan melalui proses otentikasi Kerberos termasuk otentikasi server dan *user*.

Pada layer ketiga dari Gambar .6 ini dapat kita lihat isi data yang dilewatkan jaringan. Isi data yang kita dapat hanya berupa *username* dari pengguna yang telah *login* tanpa menyertakan *password* pengguna. Hal ini berarti sistem ini cukup aman karena tidak mengirimkan *password* pada jaringan ketika proses otentikasi, sehingga *password* tidak mudah untuk dicuri.

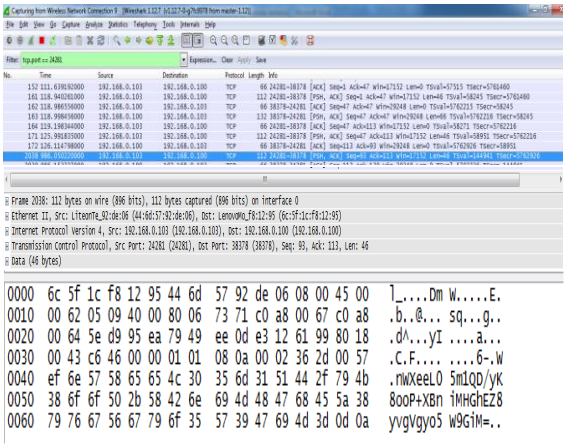


Gambar .6. Tampilan *Sniffing* Kerberos dengan *Wireshark*

D. Hasil Analisa Pengujian enkripsi AES

Proses enkripsi dan dekripsi AES terjadi pada sisi aplikasi yang ada di *user*. Adapaun hasil proses enkripsi bisa kita lihat pada server *instant message*.

Pada percobaaan ini telah diujikan pengiriman pesan dengan mengirimkan teks asli "a" pada sistem aplikasi *instant message*.



Gambar .7. Tampilan *Sniffing* AES menggunakan *Wireshark*

Gambar .7. merupakan gambar hasil *sniffing* paket pengiriman pesan yang terjadi pada *instant message*. Pada layer satu Gambar .7. merupakan gambaran informasi data yang melewati jarring, pada layer ketiga dapat dilihat isi dari data yang terkirim, berupa pesan yang telah dienkripsi dengan AES dan telah dikodekan dengan kode base64 sebagai kode standard pengiriman pesan.

V. KESIMPULAN

Kesimpulan yang didapat dalam penelitian ini ada beberpa poin penting, yaitu:

1. Aplikasi *instant message* dapat berjalan dengan baik pada sistem operasi android *jelly bean* dan versi setelahnya. Server *instant message* berjalan dengan baik pada sistem operasi *windows 7*, begitu pula server Kerberos yang berjalan dengan baik pada sistem operasi linux ubuntu.
2. Sistem keamanan *instant message* akan berjalan dengan baik apabila koneksi jaringan internet stabil.
3. Input yang diperlukan dalam sistem keamanan *instant message* ini adalah pesan teks dan file login. Jumlah karakter maksimal yang dibutuhkan untuk username dan password masing-masing adalah 30 karakter, panjang file pesan maksimal adalah 256 word.
4. Dengan adanya penerapan dua algoritma yaitu Kerberos pada otentikasi *login* dan AES pada enkripsi teks, maka diperoleh sebuah optimasi keamanan pada sebuah aplikasi *instant message* yang dibuat.
5. Proses otentikasi *login* menggunakan Kerberos cenderung lebih lama bila dibandingkan dengan sistem otentikasi *login* tanpa Kerberos, sistem otentikasi dengan Kerberos membutuhkan waktu login selama 1 menit untuk login.
6. Sistem keamanan *instant message* pada penelitian ini dianggap aman dibuktikan dengan telah diuji menggunakan teknik *brute force* dan teknik *sniffing* pada jaringan *instant message*.
7. Kerberos terbukti memiliki otentikasi yang sangat bagus, prosentase keberhasilannya adalah 95%, yang diperoleh dari persentase data uji tingkat keberhasilan dan error hasil pengujian.
8. AES-256 terbukti memiliki enkripsi yang sangat kuat dengan 14 putaran kuncinya, dan terbukti tidak mudah untuk didekrip tanpa menggunakan kunci yang sama dengan enkripsinya.

DAFTAR PUSTAKA

[1] Viewz. Instant Messaging Guide. http://www.viewz.com/features/imguide.shtml. Diakses pada tanggal 12 Oktober 2013.

[2] Christian, I. Sistem Otentikasi Keberos Pada Jaringan Komputer ITB.http://www.budi.insan.co.id/courses/ec5010/projects/ivanchristian-report.pdf, 2004.

[3] Jaganadan P. 2005.Securing text chat sistem using Kerberos authentication.Universiti Teknologi Malaysia. Malaysia.

[4] Alfred, J. Menezes, P. van Oorschot, S. Vanstone. 1996. *Handbook of Applied Cryptography*. CRC Press, USA.

[5] Munir, Rinaldi. 2006. Kriptografi. Penerbit Informatika. Bandung.

[6] Adhi, J. S. 2005. Kriptografi dengan Algoritma AES untuk Penyandian Data.*Skripsi*. Universitas Kristen Duta Wacana. Yogyakarta.

[7] Avinash K. 2014. The Advanced Encryption Standard.Purdue University.

[8] Ariyus, Doni. 2006. Kriptografi Keamanan Data dan Komunikasi.Yogyakarta.Penerbit : Graha Ilmu.

[9] *Fileperms. Vulnerability whatsapp. Guide* . http://www.fileperms.org. Diakses pada tanggal 13 Oktober 2013.

[10] *Firstsight.* Line chat Vulnerability. http://firstsight.me. Diakses pada tanggal 5 November 2014.

[11] Kurniawan , Yusuf. 2003. AES (Advanced encryption standard)**.**Jurnal Informatika. Universitas Pasundan.

[12] Schneier, B. 1996.Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C. John Wiley & Sons, Inc

[13] Stallings, William 2011. Cryptography and Network Security : Principles and Practice (5th Edition). Prentice Hall. United States of America.

[14] Yuniati, V. and Indriyana, G. 2009.Enkripsi dan Dekripsi dengan Algoritma AES 256 Untuk Semua Jenis File. *Jurnal Informatika ITB.* Bandung