

Pengamanan Komunikasi Suara Melalui Internet Pada Telepon Seluler dengan Algoritma Tea Pada Platform Android

Denver¹, Rinaldi Munir²

School of Electrical Engineering and Informatics, Institute Technology of Bandung,
10th Ganeca Street, Bandung, Indonesia.
deng37@gmail.com¹, rinaldi@informatika.org²

Abstrak— Komunikasi suara pada telepon seluler melalui internet sudah mulai berkembang. Keuntungan yang diperoleh ialah rendahnya biaya yang dibutuhkan. Tetapi keuntungan ini perlu dibayar dengan lemahnya proteksi. Oleh karena itu, dibuatlah suatu pengaman untuk mengamankan komunikasi suara tersebut.

Dalam tugas akhir ini, dibuat sebuah aplikasi pengamanan komunikasi suara dengan menggunakan *Tiny Encryption Algorithm* (TEA). Aplikasi yang dibangun ditujukan untuk telepon seluler dengan *platform* Android. Sebelum aplikasi dibuat, akan dilakukan analisis terkait masalah yang muncul dan perancangan solusinya. Kemudian dilakukan analisis terkait kebutuhan sistem dan perancangan perangkat lunak. Pengujian dilakukan pada dua perangkat Android. Pengujian ini meliputi tiga aspek yaitu pengujian *delay* telepon dengan enkripsi dekripsi, pengujian *delay* telepon tanpa enkripsi dekripsi, dan pengujian hasil enkripsi.

Hasil pengujian memperlihatkan bahwa *delay* lebih besar dari batas yang ditentukan. Tetapi hal ini dapat ditoleransi karena nilai lebih yang dimilikinya. Nilai lebih ini adalah pengamanan yang dilakukan pada komunikasi yang berjalan.

Kata Kunci — komunikasi suara, telepon seluler, internet, TEA, enkripsi

I. LATAR BELAKANG

Pada jaman sekarang ini, orang-orang semakin mudah melakukan komunikasi suara. Dengan adanya telepon seluler, komunikasi suara dapat dibangun dengan *mobile*. Komunikasi suara ini sudah berkembang memasuki dunia internet. Adapun keuntungan komunikasi suara pada jaringan internet adalah biaya yang murah ketimbang komunikasi suara melalui *Public Switched Telephone Network* (PSTN).

Karena komunikasi ini berjalan dalam jaringan internet, tingkat keamanan menjadi rendah. Penyadapan lebih mudah dilakukan. Dalam melakukan penyadapan ini biasanya memiliki tujuan untuk mendapatkan informasi yang tidak seharusnya didapatkan oleh penyadap ini. Dengan didapatnya informasi ini, maka pihak yang melakukan komunikasi ini akan merasa dirugikan. Oleh karena itu, perlu dilakukan pengamanan terhadap komunikasi suara tersebut.

Pada tugas akhir ini akan dibuat sebuah aplikasi yang dapat melakukan komunikasi suara dengan paket-paket suara yang dikirimkan, diamankan terlebih dahulu. Aplikasi ini akan dibangun pada telepon seluler dengan *platform* Android. Android merupakan salah satu sistem operasi pada *smartphone*. Berbagai perusahaan telepon seluler menggunakan Android sebagai sistem operasinya.

Pengamanan yang dilakukan pada komunikasi suara dapat bermacam-macam. Metode jaman dahulu yakni dengan mengacaukan sinyal masukan seperti pada penelitian [5]. Adapun metode pengamanan yang diterapkan dengan metode kriptografi modern. Pengamanan akan dilakukan pada masing-masing bit masukan, melewati serangkaian pengamanan, kemudian menghasilkan keluaran yang sama sekali berbeda dengan masukan.

Algoritma yang dipakai untuk pengamanan komunikasi suara di dalam Tugas Akhir ini adalah algoritma *Tiny Encryption Algorithm* (TEA). Algoritma TEA merupakan algoritma *cipher* blok dengan ciri khas berupa kode yang tidak panjang tetapi kuat, dan cepat. Kekuatan algoritma ini sekomples *Data Encryption Standard* (DES) dan kesederhanaannya membuat algoritma ini dapat ditranslasikan pada berbagai bahasa serta digunakan pada berbagai macam alat komputasi. [9].

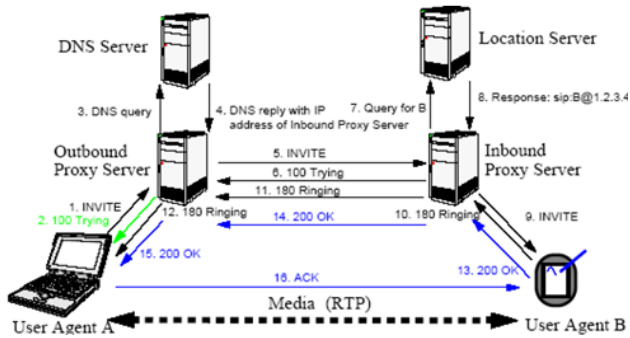
II. DASAR TEORI

A. Komunikasi Suara pada Internet

Komunikasi suara yang dibangun pada internet membutuhkan dua tahapan. Tahapan yang pertama berupa *call setup* yakni pengaturan telepon yang mencakup semua detail terkait komunikasi agar komunikasi dapat berjalan. Tahapan kedua yang berupa pengiriman data melalui internet, menggunakan protokol SIP yang beroperasi pada layer aplikasi. Arsitektur dari SIP adalah *user agent* dan *server*. *User agent* merupakan aktor dari sistem dan memiliki dua subsistem yaitu *user agent client* (UAC) dan *user agent server* (UAS). UAC ini berfungsi untuk membangkitkan *request* dan UAS berfungsi untuk merespon *request*. Sedangkan *server* ini sendiri memiliki empat bagian menurut [2] yakni *proxy server*, *redirect server*, *registrat server*, dan *location server*.

Pada Gambar II.1 [8] digambarkan percakapan menggunakan protokol SIP antara *user agent* A dan B. Masing-

masing *user agent* memiliki *proxy server* dan *DNS server*. Kedua *proxy server* meminta alamat dari *DNS server*. Alamat ini akan dipertukarkan untuk kemudian dipakai masing-masing *user agent* untuk mencapai *user agent* yang lain. Komunikasi selanjutnya akan berjalan pada protokol RTP.



Gambar II.1. Skenario percakapan yang dibangun diatas SIP

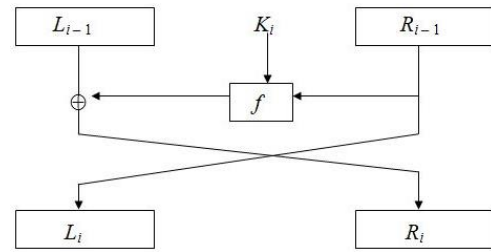
B. Algoritma Kriptografi Modern

Ide dasar algoritma kriptografi modern ialah berasal dari algoritma klasik. Algoritma kriptografi klasik memiliki 2 ide dasar yakni *cipher* substitusi dan *cipher* tranposisi/permutasi. *Cipher* substitusi melakukan pertukaran masing-masing karakter dengan karakter-karakter lain yang berbeda. *Cipher* tranposisi melakukan pengubahan posisi yang terdapat pada karakter-karakter. Dengan menggabungkan substitusi dan tranposisi pada bit-bit karakter maka muncullah algoritma kriptografi modern ini [1].

Algoritma kriptografi modern memproses baik *plaintext*, *ciphertext*, maupun kunci dalam rangkaian bit. Operasi yang seringkali digunakan adalah operasi bit XOR (*exclusive OR*). Operasi bit ini memiliki maksud semua data masukan akan diproses dalam bentuk bit *biner* (nol dan satu). Dokumen seperti makalah atau laporan, dalam media digital, telah mempunyai representasi *biner*. Hanya saja yang tertampil dalam representasi karakter, yang dapat dibaca. Setelah melakukan enkripsi dalam representasi *biner* maka hasil enkripsi/*ciphertext* tidak dapat terbaca lagi, menjadi kacau.

Cipher blok merupakan algoritma modern yang beroperasi pada sejumlah blok bit yang telah ditentukan. Masing-masing blok bit tersebut akan diproses sendiri-sendiri. *Cipher* blok ini menyajikan pengamanan lebih baik ketimbang *cipher* aliran, tetapi membutuhkan waktu lebih banyak untuk memroses ketimbang dengan *cipher* aliran. Hal ini disebabkan *cipher* blok harus menunggu satu blok terpenuhi untuk dapat diproses. Skema *cipher* blok dapat dilihat pada gambar II.7 [6].

Kunci dari *cipher* blok merupakan kunci yang tetap, *cipher* blok ini tidak seperti *cipher* aliran memiliki *keystream generator*, penghasil kunci untuk setiap masukan. Kunci yang dihasilkanpun berbeda-beda. Kunci pada *cipher* blok memiliki panjang yang sama dengan panjang blok itu sendiri.

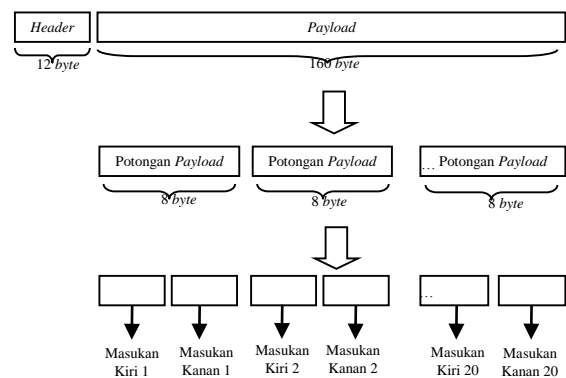


Gambar II.2. Konsep *cipher* blok dengan fungsi *round*

Dalam perancangannya, terdapat lima prinsip yang digunakan yakni *confusion* – *diffusion*, *cipher* berulang, *substitution box*, kunci lemah, dan jaringan *Feistel*. Jaringan *Feistel* dipakai pada berbagai macam algoritma karena bersifat *reversible*. *Reversible* disini mengandung makna proses enkripsi dan dekripsi hampir serupa, hanya berkebalikan. Hal ini mengakibatkan jaringan *Feistel* ini banyak digunakan pada berbagai algoritma. Jaringan *Feistel* merupakan perulangan *cipher* dimana didalamnya terdapat fungsi yang dinamakan fungsi *round*. Perulangan sebanyak satu kali ditampilkan pada gambar II.2.

C. Algoritma TEA

Algoritma TEA merupakan salah satu algoritma *cipher* blok. Algoritma ini diajukan pada *Fast Software Encryption*, dan telah mengalami beberapa perubahan. Mulai dari TEA versi asli, kemudian dikembangkan menjadi *extended TEA* (XTEA), dan perubahan terakhir menjadi *corrected* blok TEA (XXTEA). Algoritma ini dibuat dengan tujuan menghasilkan suatu algoritma dengan penggunaan memori sekecil-kecilnya untuk mendapat tingkat performansi dan pengamanan yang baik. Dengan kesederhanaannya, diharapkan algoritma TEA ini mampu diimplementasikan pada sebagian besar alat komputasi [9].



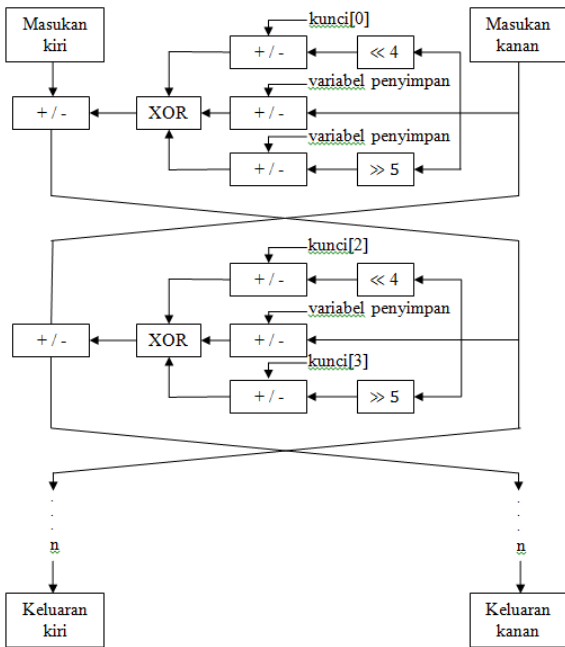
Gambar II.3 Skema Pemecahan Paket menjadi Masukan Enkripsi/Dekripsi

III. PEMBAHASAN

A. Analisis Masalah

Berikut merupakan berbagai permasalahan yang perlu diperhatikan:

1. Penggunaan API dari Android memberikan kemudahan dalam membangun komunikasi melalui protokol internet. Dalam Tugas Akhir akan dilakukan pengamanan pada bit-bit paket suara yang akan dikirim. Tetapi diperlukan suatu pengaksesan pada paket-paket yang akan dikirim. API Android tidak dapat diakses sehingga dapat dipilih aplikasi Sipdroid [4]. Aplikasi ini membangun sendiri komunikasi suara melalui internet. Sehingga penyisipan enkripsi dan dekripsi dapat dilakukan.
2. Penyedia layanan SIP dibagi menjadi berbayar dan tidak berbayar. Untuk mencapai tujuan nilai ekonomis yang rendah, dipilih layanan yang tidak berbayar. Tetapi layanan SIP yang tidak berbayar ini ada berbagai kendala yang bisa terjadi seperti adanya batasan durasi telepon, *server* yang suka mati, hingga proses pendaftaran yang menyulitkan pengguna. SIP Linphone dipilih karena layanan SIP tidak berbayar ini tidak memiliki kendala yang disebutkan.
3. Dalam membangun komunikasi melalui internet, terdapat dua jenis sambungan yakni *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP). TCP cenderung lebih lama dalam membangun koneksi, tetapi koneksi lebih terjamin pada saat komunikasi sudah berjalan. TCP dipilih karena jaminan komunikasinya.
4. Berdasarkan implementasi algoritma TEA, SNOW, dan KASUMI, didapatkan bahwa algoritma SNOW dan KASUMI tidak dapat dipakai. Kedua algoritma ini menghasilkan lama waktu enkripsi dan dekripsi yang terlalu besar. Lama waktu yang dihasilkan dari proses enkripsi dan dekripsi tidak boleh terlalu besar, bila terlalu besar maka paket-paket suara yang ada menjadi pecah, tidak dalam satu kesatuan dan komunikasi tidak dapat dilakukan. Dengan algoritma TEA lama waktu enkripsi dekripsi jauh lebih kecil dan komunikasi suara dapat berjalan.
5. Kompresi paket dapat dilakukan secara *lossy* maupun *lossless*. Paket-paket yang hilang dapat menyulitkan proses enkripsi dan dekripsi karena urutannya menjadi tidak sesuai. Penggunaan kompresi *lossless* dirasa lebih tepat.
6. Sekalipun metode pengompresian *lossless*, tidak dapat menjamin semua paket sampai. Hal utama yang dapat mempengaruhi ialah masalah konektivitas internet yang lambat. Karena dimungkinkan adanya paket hilang, diperlukan penanganan pada saat dekripsi dilakukan.
7. Kunci yang menjadi masukan pada saat enkripsi dan dekripsi dilakukan, perlu ditangani apabila kunci masukan pengguna tidak sesuai ukuran yang diperlukan. Fungsi MD5 menghasilkan keluaran yang sesuai sebagai masukan kunci enkripsi dan dekripsi.



Gambar II.4. Skema fungsi *round* pada iterasi pertama sampai iterasi terakhir

Skema pada gambar II.3 merupakan tahapan pemecahan paket sampai menjadi masukan untuk enkripsi atau dekripsi. Masukan ini kemudian akan diproses oleh algoritma TEA seperti pada gambar II.4. Fungsi ini dapat berupa enkripsi atau dekripsi. Hanya berbeda pada operasi penambahan dan pengurangan saja. Operasi penambahan bekerja pada enkripsi, sedangkan operasi pengurangan bekerja pada dekripsi.

D. Algoritma SNOW dan KASUMI

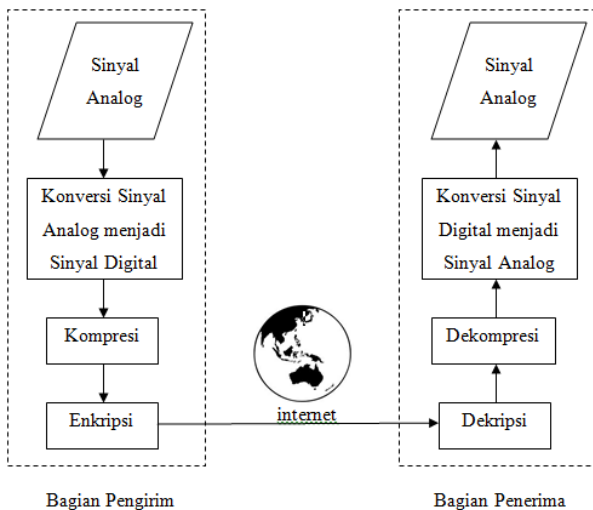
Algoritma SNOW merupakan salah satu algoritma *cipher* aliran. Algoritma ini diajukan pada *New European Schemes for Signatures, Integrity, and Encryption* (NESSIE) dan algoritma ini telah mengalami beberapa perubahan. Mulai dari SNOW 1.0, SNOW 2.0, sampai dengan SNOW 3G. Algoritma SNOW yang dipakai disini adalah SNOW 3G.

Algoritma SNOW 3G terdiri dari dua bagian yakni *Linear Feedback Shift Register* (LFSR) dan *Finite State Machine* (FSM). LFSR merupakan *shift register* yang masukannya adalah hasil keluaran dari fungsi dari keadaan sebelumnya. Terkadang keluaran LFSR ini dikombinasikan dengan XOR. Proses enkripsi dan dekripsi yang dilakukan pada algoritma ini serupa [7].

Algoritma KASUMI merupakan algoritma *cipher* blok. Algoritma ini dibuat oleh 3rd *Generation Partnership Project* (3GPP) untuk mengamankan komunikasi pada telepon seluler. Struktur algoritma KASUMI adalah jaringan Feistel dengan masukan sebesar 64 bit dan keluaran sebesar 64 bit. Kunci yang digunakan sebesar 128 bit. Algoritma KASUMI terdiri dari beberapa sub fungsi-fungsi yang digunakan bersama-sama dengan beberapa sub kunci-kunci [3].

B. Rancangan Perangkat Lunak dan Rancangan Solusi

Pengamanan komunikasi yang dibangun memiliki dua bagian utama yakni bagian pengirim dan bagian penerima. Kedua bagian ini dapat bekerja pada waktu bersamaan, ditunjukkan pada gambar III.1.



Gambar III.1: Arsitektur Aplikasi yang Dibangun

Aplikasi yang dibangun dapat berjalan dua arah. Pada saat yang bersamaan aplikasi dapat melakukan enkripsi dan dekripsi bila terdapat paket yang keluar dan masuk. Berikut merupakan berbagai perancangan solusi yang dibangun:

1. Penanganan Pesan untuk Enkripsi Dekripsi. Masukan algoritma TEA berukuran delapan *byte*. Paket yang diterima atau dikirim berisikan *payload* berukuran 160 *byte* menjadi dua puluh bagian, masing-masing berukuran delapan *byte*.
2. Mekanisme Penanganan Paket Hilang. Paket hilang membuat proses dekripsi dapat menjadi kacau, oleh karena itu dilakukan dekripsi buatan sejumlah paket yang hilang sehingga *counter* memiliki nilai yang sama dengan paket yang diterima. Dekripsi buatan adalah proses dekripsi dengan paket kosong.
3. Mekanisme Penanganan Kunci. Kunci masukan algoritma TEA yakni sebesar 128 bit. Bila masukan pengguna sebesar 128 bit, maka masukan pengguna akan dijadikan masukan kunci. Akan tetapi bila masukan pengguna tidak sama dengan 128 bit, maka masukan pengguna akan dimasukkan ke dalam fungsi hash MD5 sehingga keluaran tepat 128 bit. Fungsi hash ini berguna dalam menghasilkan perubahan pada tiap bit.

IV. PENGUJIAN

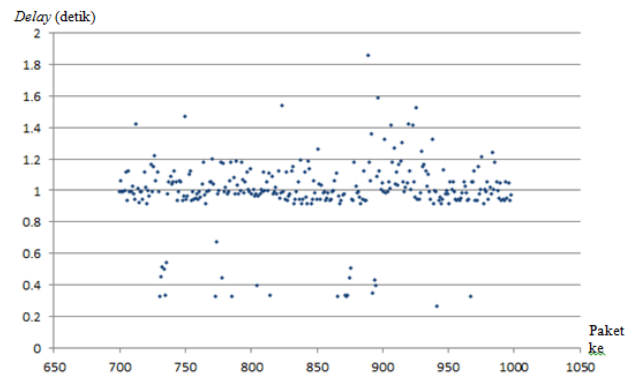
Pengujian dibagi menjadi tiga bagian yakni pengujian *delay* dengan enkripsi dekripsi, pengujian *delay* tanpa enkripsi dekripsi, dan pengujian hasil enkripsi. Pengujian ini dilakukan pada dua telepon seluler Android yakni Samsung Galaxy Tab

P3100 dan SNexian Mi320. Komputer jinjing Asus U36S digunakan untuk menangkap paket-paket yang dikirim.

A. Pengujian Delay dengan Enkripsi dan Dekripsi

Pengujian *delay* dengan enkripsi dan dekripsi dilakukan pada kedua perangkat dengan menyalakan Wi-Fi (untuk SNexian Mi320) dan menyalakan *packet data* (untuk Samsung P3100), menjalankan aplikasi, melakukan pengaturan ID SIP, menunggu sampai aplikasi terhubung ke *server* (ditandai dengan berubahnya bulatan kuning pada jendela notifikasi menjadi berwarna hijau), dan memasukkan kunci. Setelah itu pada Samsung P3100 (*sip:byupayoy@sip.linphone.org*) diisi alamat SIP SNexian (*sip:deng37@sip.linphone.org*) dan panggilan dilakukan. Untuk Samsung P3100 perlu mencentang bagian “Use 3G” karena jaringan yang digunakan adalah internet dari kartu Telkomsel Halo.

Komputer jinjing ini menjalankan program Eclipse untuk mencatat lama waktu yang dibutuhkan dari proses enkripsi, pengiriman, sampai dekripsi. Sampel data yang digunakan sebanyak tiga ratus paket, dari paket ke tujuh ratus sampai paket ke seribut. Persebaran data yang digunakan dapat dilihat pada gambar V.4. Dari ketiga ratus paket tersebut, waktu *delay* tersebut dirata-ratakan dan didapatkan hasil tertera pada tabel V.2.



Gambar V.4: Grafik Persebaran Data Delay dengan Enkripsi dan Dekripsi

Table V.2: Tabel Hasil Pengujian Delay dengan Enkripsi dan Dekripsi

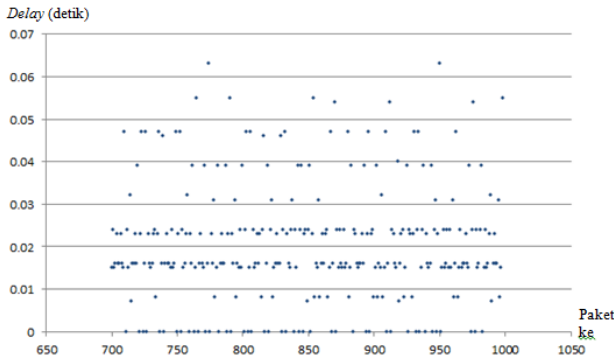
Kasus Uji	Rata-rata Delay (mili detik)
Pengukuran <i>delay</i> dengan enkripsi dan dekripsi	989,686

B. Pengujian Delay tanpa Enkripsi dan Dekripsi

Pengujian kedua ini mengukur *delay* yang dihasilkan tanpa adanya proses enkripsi dan dekripsi. Skema pengujian yang digunakan sama dengan pengujian pada V.2.3.1, hanya saja terdapat langkah yang berbeda. Kunci tidak perlu dimasukkan dan tombol “Encryption” ditekan sehingga status enkripsi menjadi “off”.

Komputer jinjing menjalankan program Eclipse untuk mencatat *delay*. Sampel data yang digunakan sebanyak tiga

ratus, dari paket ke tujuh ratus sampai paket ke seribu. Persebaran data dapat dilihat pada gambar V.5. Dari ketiga ratus paket tersebut, *delay* dirata-ratakan dan didapatkan hasil tertera pada tabel V.3.



Gambar V.5: Grafik Persebaran Data *Delay* tanpa Enkripsi dan Dekripsi

Table V.3: Tabel Hasil Pengujian *Delay* tanpa Enkripsi dan Dekripsi

Kasus Uji	Rata-rata <i>Delay</i> (mili detik)
Pengukuran <i>delay</i> tanpa enkripsi dan dekripsi	20,013

C. Pengujian Hasil Enkripsi

Pengujian terakhir, menentukan apakah *ciphertext* dengan plaintexts merupakan data yang berbeda. Pengujian ini dilakukan dengan menggunakan program Wireshark, digunakan untuk menangkap paket yang dikirim. Paket yang ditangkap Wireshark ini dibandingkan dengan paket belum dienkripsi dan belum dikirim, dicatat pada program Eclipse. Hasil penangkapan paket untuk nomor paket ke dua ratus ditampilkan pada tabel V.4.

Table V.4: Tabel Perbandingan Paket Setelah Dikirim dan Sebelum Enkripsi

Paket Setelah Dikirim	ce 38 42 8f 0f 01 55 01 ce 38 42 8f 0f 01 55 01 18 f3 02 d1 5d d3 24 77 0e fa 3e 36 d3 e7 75 6e ec 29 8d 0e 13 7b 7b 19 65 e6 14 23 d3 17 a3 7a c0 12 4d 86 d1 33 65 f5 28 c4 3c ba cd 49 b8 b1 ec 11 3c f7 58 f0 26 3e 73 3e e6 11 a3 f0 d2 5c 60 f7 43 63 9c 4f 2d 8a d3 97 11 62 16 35 21 4e ce a3 5d 1c 39 d5 d7 06 fe 49 71 3b da 7c 10 e0 a7 c6 49 cd 6e dc f0 b6 d8 b2 39 6d c0 f9 0b 75 59 4b 4b 7d eb 23 fe 6b ba 65 f4 ea 3a da ff f9 13 e4 6b 8b a5 ff fd ae b2 14 c3 07 39 d0 3a d9
Paket Sebelum Enkripsi	d5 55 d5 d5 54 55 55 54 d5 55 55 55 d5 d5 d5 d5 55 55 d5 d5 d5 55 d5 d4 d4 55 55 55 55 55 55 55 55 55 55 55 55 d5 d5 55 55 d5 d5 d5 55 d5 55 55 55 54 54 55 55 54 55 55 d5 d5 d5 d4 d5 d5 d5 d5 55 55 d5 55 54 55 55 d5 55 55 55 55 d5 55 d5 55 54 d5 d5 55 d5 d5 d5 d5 d5 d5 d5 55 55 55 54 54 54 55 55 55 55 55 d5 d5 d4 d4 d4 d5 d5 d5 55 55 54 55 55 54 54 55 55 54 55 55 d5 55 55 55 55 d5 d5 55 d5 55 54 55 d5 d5 55 55 d5 55 55 55

D. Analisis Hasil Pengujian

Menurut ITU tertera pada spesifikasi G.114 (International Telecommunication Union, 1993), direkomendasikan *delay*

sebesar 150 mili detik untuk kualitas tinggi yang *real time*. Adapun untuk panggilan internasional memiliki rekomendasi *delay* sebesar 300 mili detik. Diambil nilai rekomendasi 300 mili detik karena *server* yang digunakan berada di luar negeri, Perancis.

Hasil pengujian didapatkan bahwa untuk menyampaikan satu paket dari mulai dari enkripsi, pengiriman paket, dan dekripsi dibutuhkan waktu 989,686 mili detik. *Delay* yang dihasilkan kurang lebih tiga kali lebih lambat ketimbang nilai rekomendasi.

Akan tetapi dalam standar tersebut (International Telecommunication Union, 1993) disebutkan bahwa bila *delay* melebihi angka rekomendasi, masih dapat diterima jika memiliki suatu nilai lebih. Nilai lebih tersebut dapat berupa kualitas yang baik ataupun jarang komunikasi terputus. Aplikasi yang dibangun memberikan nilai lebih berupa keamanan pada komunikasinya. Adapun pada saat pengujian belum pernah mengalami terputusnya komunikasi, walaupun koneksi internet pada saat tersebut relatif buruk.

Delay yang dihasilkan dengan dan tanpa enkripsi dekripsi memiliki selisih waktu sebesar 969,672 mili detik. Hal ini dapat dilihat bahwa peran enkripsi dan dekripsi dalam menghasilkan *delay* sangat besar. *Delay* yang berasal dari enkripsi dekripsi ini dapat berkurang nilainya bila menggunakan perangkat Android dengan kemampuan prosesor lebih baik.

Delay yang dihasilkan tidak boleh terlalu besar. Bila *delay* memiliki nilai yang sangat besar maka suara tidak dapat didengar lagi. Suara yang tidak terdengar ini diakibatkan paket-paket yang diterima menjadi pecah, bukan lagi merupakan satu kesatuan. Suara dapat terdengar bila jarak antar paket cukup berdekatan, bila jarak antar paket jauh, maka yang terdengar hanyalah potongan suara. Potongan-potongan suara ini bila didengar akan menyerupai suara terenkripsi yang tidak didekripsi, bisings.

Dari pengujian dengan telepon terenkripsi, dapat dilihat bahwa aplikasi ini dapat melakukan komunikasi antara kedua perangkat dimana salah satu perangkatnya memiliki spesifikasi rendah. Telepon seluler SNexian Mi320 merupakan salah satu telepon seluler Android Gingerbread dengan spesifikasi rendah. Tapi aplikasi dapat berjalan diatas telepon seluler ini. Hal ini dikarenakan pemilihan algoritma TEA yang tepat. Algoritma ini ringan sehingga *processor* SNexian Mi320 yang tidak hebat sekalipun dapat berjalan dengan cepat.

Pengujian *delay* dengan menggunakan kartu Telkomsel Halo dapat dilakukan dengan syarat sinyal yang didapat minimal pada rentang 3G keatas. Hal ini dapat dilakukan dengan melakukan telepon diluar bangunan berbeton. Bangunan bertingkat sekarang memiliki beton-beton yang dapat menghalangi sinyal masuk. Adapun bangunan bertingkat yang modern memiliki *transmitter* sehingga sinyal yang didapat tetap baik.

Koneksi *Asymmetric Digital Subscriber Line* (ADSL) Telkom Speedy pada pengujian baik untuk digunakan. Koneksi ini menggunakan sambungan telepon PSTN. Sambungan ini lebih stabil ketimbang menggunakan jaringan telepon seluler. Telepon PSTN tidak mengenal adanya derajat sinyal yang diterima seperti halnya telepon seluler.

Pengujian langsung untuk mengetahui bahwa paket benar terenkripsi dilakukan dengan mengatur salah satu perangkat pada mode terenkripsi, dan perangkat satu lagi dalam mode tidak terenkripsi. Hasil keluaran yang didengar berupa suara bising. Suara bising ini merupakan suara asli yang terenkripsi tapi tidak didekripsi, karena pada penerima mode enkripsi dimatikan. Suara bising juga akan dihasilkan bila kedua belah pihak memasukkan kunci yang berbeda.

V. KESIMPULAN

Pada pengerjaan Tugas Akhir ini, dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Pengamanan dapat dilakukan pada komunikasi suara yang dibangun diatas jaringan internet, algoritma TEA sebagai pengamanannya, dan telepon seluler Android dengan versi minimal 2.3 (Android Gingerbread).

2. Dengan algoritma TEA, pengamanan yang dilakukan tidak merusak jalannya komunikasi suara.

3. *Delay* yang dihasilkan dari komunikasi suara dengan enkripsi dan dekripsi memiliki yakni sebesar 989,686 mili detik. *Delay* ini melebihi batas rekomendasi (300 mili detik), tetapi masih dapat ditoleransi karena nilai lebih yang diberikan berupa pengamanan yang dilakukan pada komunikasi suaranya.

4. Perbedaan *delay* antara komunikasi suara dengan dan tanpa enkripsi dekripsi adalah 969,672 mili detik. Angka ini mendekati *delay* komunikasi suara dengan enkripsi dan dekripsi yang berarti proses enkripsi dan dekripsi mempengaruhi hampir

semua *delay* yang dihasilkan.

5. Data suara yang dienkrpsi benar aman karena antara data yang dikirim (data yang telah dienkrpsi) dengan data sebelum dienkrpsi adalah berbeda.

REFERENSI

- [1] [1] Bishop, David (2003). *Introduction to Cryptography with Javtm Applets*. Massachusetts, Jones and Bartlett Publishers.
- [2] [2] Carlson, Ian, Charles Avila (2004). *Voice over IP (VoIP)/SIP Infrastructure Considerations for the Interaction Center Platform*. Interactive Intelligence.
- [3] [3] ETSI/SAGE (1999). KASUMI Specification: *Specification of the 3GPP Confidentiality and Integrity Algorithms Document 2*. Sophia Antipolis, France.
- [4] [4] Google Code (2009): Free SIP/VoIP Client for Android, <https://code.google.com/p/sipdroid/>, *download(diturunkan/diunduh)* pada 27 April 2013.
- [5] [5] Islam, Saad, Fatima Ajmal, Salman Ali, Jawad Zahid, Adnan Rashdi (2009). *Secure End-to-End Communication over GSM and PSTN Networks*. IEEE International Conference, National University of Sciences and Technology, Pakistan.
- [6] [6] Munir, Rinaldi (2005). *Kriptografi*. Diktat Kuliah IF5054, Institut Teknologi Bandung.
- [7] [7] Orhanau, Ghizlane, Said El Hajji, Youssef Bentaleb (2010). *SNOW 3G Stream Cipher Operation and Complexity Study*. Contemporary Engineering Sciences, Vol. 3, 2010, 97-111, Universite Mohammed V Agdal, Maroc.
- [8] [8] Tong, Hoang Anh (2005). *SIP-Based VoIP Service – Architecture and Comparison*. University of Stuttgart.
- [9] [9] Wheeler, David J, Roger Needham (1994). *TEA, a tiny encryption algorithm*. Fast Software Encryption, Leuven, Belgium.