Analisis Sistem Keamanan Menggunakan Kriptografi pada Aplikasi Skype

Hapsari Tilawah 13509027[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13509027@std.stei.itb.ac.id

***Abstract*—Skype menggunakan algoritma enkripsi standar yang diakui dan diterima secara internasional seperti algoritma cipher blok AES, kriptosistem kunci publik RSA, skema *signature padding* ISO 9796-2, fungsi hash SHA-1, dan cipher stream RC4. Makalah ini bertujuan untuk mempelajari algoritma enkripsi yang digunakan Skype dan mencoba untuk melakukan dekripsi data *stream* Skype. Data *stream* Skype dapat diperoleh menggunakan *tool* Wireshark melalui *port* tertentu. Namun data *stream* tersebut tidak berhasil didekripsi karena sulit untuk menebak 2^{32} kunci RC4.**

***Index Terms*—Skype, kriptografi, serangan, AES, RSA, RC4, Wireshark**

I. PENDAHULUAN

Internet, sebagaimana setiap jaringan apapun, dapat dipantau oleh penjahat dan *hacker* di sejumlah titik. Hal ini menjadi salah satu alasan mengapa email dan banyak program internet *chatting* tidak aman. Oleh karena ada begitu banyak cara untuk memantau komunikasi oleh orang yang tidak dikenal, pengguna harus mengambil langkah positif untuk melindungi diri dari pihak ketiga berbahaya.

Enkripsi adalah proses untuk melakukan konversi informasi dengan menggunakan prinsip matematika sedemikian rupa sehingga hanya dapat dibaca oleh penerima yang dimaksud setelah dikonversi menjadi informasi kembali. Banyak jenis teknik enkripsi telah dikembangkan selama berabad-abad. Proses ini disebut enkripsi dan dekripsi yang merupakan bagian dari disiplin keamanan yang disebut kriptografi.

Skype adalah sistem *Voice over Internet Protocol* (VoIP) yang dikembangkan oleh Skype Technologies S.A. Skype adalah jaringan *peer-to-peer* dimana panggilan suara melewati internet daripada melalui jaringan tujuan yang khusus. Skype memberikan layanan yang memungkinkan pengguna untuk berkomunikasi dengan *peers* lewat suara dengan menggunakan mikrofon, video dengan menggunakan webcam, dan pesan instan melalui Internet. Panggilan ke pengguna lain dalam layanan Skype adalah bebas biaya, sedangkan panggilan ke telepon dan ponsel dibebankan biaya melalui sistem berbasis rekening debit pengguna. Skype juga menjadi populer untuk fitur tambahannya, termasuk transfer file, dan konferensi video.

Skype menggunakan kriptografi secara luas untuk mengotentikasi identitas pengguna dan server dan untuk melindungi konten yang ditransmisikan melalui jaringan *peer-to-peer* agar tidak jatuh ke tangan *hacker* dan penjahat. Sistem kriptografi yang direkayasa untuk tujuan ini telah dirancang dengan baik dan benar dilaksanakan.

Skype menggunakan algoritma enkripsi standar yang diakui dan diterima secara internasional serta telah tahan uji selama bertahun-tahun dari analisis dan serangan. Skype hanya menggunakan kriptografi primitif standar untuk memenuhi kebutuhannya, yang merupakan pendekatan rekayasa suara. Primitif-primitif ini termasuk algoritma cipher blok AES, kriptosistem kunci publik RSA, skema *signature padding* ISO 9796-2, fungsi hash SHA-1, dan cipher stream RC4.

Skype mengoperasikan otoritas sertifikat untuk nama pengguna dan otorisasi. *Digital signature* yang dibuat oleh otoritas ini adalah dasar bagi identitas di Skype. Node Skype yang masuk ke sebuah sesi (*session)* melakukan verifikasi identitas dari *peer* mereka. Tidak mungkin bagi penyerang untuk menipu identitas Skype pada atau di bawah *session layer*. Dengan demikian, Skype memastikan terjaganya privasi serta integritas data yang dikirim dari atau ke kontak penggunanya.

II. DASAR TEORI

A. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) adalah sebuah spesifikasi untuk enkripsi data elektronik yang didirikan oleh U.S. National Institute of Standards and Technology (NIST) pada tahun 2001. Berdasarkan cipher Rijndael yang dikembangkan oleh dua kriptografer Belgia, Joan Daemen dan Vincent Rijmen, yang mengajukan proposal yang dievaluasi oleh NIST selama proses seleksi AES. Setiap putaran mengunakan kunci internal yang berbeda (disebut *round key*). Enciphering dalam AES melibatkan operasi substitusi dan permutasi. Karena AES menetapkan panjang kunci adalah 128, 192, dan 256, maka dikenal AES-128, AES-192, dan AES-256. Garis besar algoritma

AES adalah sebagai berikut:

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*.
3. *Final round*: proses untuk putaran terakhir:
    a. SubBytes
    b. ShiftRows
    c. AddRoundKey

B. RSA

RSA adalah sebuah algoritma pada enkripsi kunci publik. RSA termasuk algoritma asimetri karena mempunyai dua kunci, yaitu kunci publik dan kunci privat. RSA ditemukan oleh tiga peneliti dari MIT (Massachussets Institute of Technology), yaitu Ron Rivest, Adi Shamir, dan Len Adleman, pada tahun 1976. RSA merupakan algoritma pertama yang cocok untuk digital signature seperti halnya ekripsi, dan salah satu yang paling maju dalam bidang kriptografi kunci publik. RSA masih digunakan secara luas dalam protokol electronic commerce, dan dipercaya dalam mengamankan dengan menggunakan kunci yang cukup panjang. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pembangkitan pasangan kunci RSA adalah sebagai berikut:

1. Pilih dua bilangan prima, a dan b (rahasia)
2. Hitung n = a b. Besaran n tidak perlu dirahasiakan.
3. Hitung $\phi(n) = (a - 1)(b - 1)$.
4. Pilih sebuah bilangan bulat untuk kunci publik, sebut namanya e, yang relatif prima terhadap $\phi(n)$.
5. Hitung kunci dekripsi, d, melalui $ed \equiv 1 \pmod{m}$ atau $d \equiv e^{-1} \bmod (\phi(n))$

Hasil dari algoritma di atas adalah kunci publik, pasangan (e, n), dan kunci privat, pasangan (d, n).

C. Fungsi Hash

Fungsi hash sering disebut sebagai fungsi satu arah (*one-way function*). Fungsi ini mengubah suatu input menjadi output, tetapi output tersebut tidak dapat dikembalikan menjadi bentuk semula. Salah satu manfaatnya adalah penggunaan sidik jari (fingerprint). Sidik jari digunakan sebagai identitas pengirim pesan. Fungsi lain adalah untuk kompresi dan message digest. Contoh algoritma fungsi ini adalah MD-5 dan SHA.

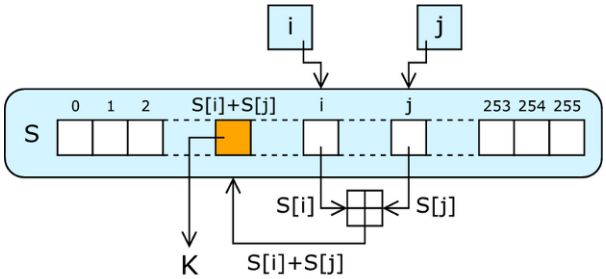
D. RC4

RC4 adalah algoritma kriptografi yang termasuk ke dalam cipher aliran (*stream cipher*). RC4 dibuat oleh Ron Rivest (1987) dari Laboratorium RSA. RC adalah singkatan dari Ron's Code. Versi lain megatakan Rivest Cipher. RC4 digunakan dalam sistem keamanan seperti protokol SSL (Secure Socket Layer), WEP (Wired Equivalent Privacy), dan WPA (Wi-fi Protect Access) untuk nirkabel. RC4 membangkitkan aliran-kunci (*key stream*) yang kemudian di-XOR-kan dengan plainteks RC4 memproses data dalam ukuran byte, bukan dalam bit. Untuk membangkitkan aliran-kunci, cipher menggunakan status internal yang terdiri dari:

- Permutasi angka 0 sampai 255 di dalam larik S0, S1, …, S255. Permutasi merupakan fungsi dari kunci U dengan panjang variabel.
- Dua buah pencacah indeks, i dan j

Algoritma RC4 adalah sebagai berikut:

1. Inisialisasi larik S: S0 = 0, S1 = 1, …, S255 = 255
2. Jika panjang kunci U < 256, lakukan *padding* sehingga panjang kunci menjadi 256 byte.
3. Lakukan pengacakan (permutasi) nilai-nilai di dalam larik S.
4. Bangkitkan aliran-kunci dan lakukan enkripsi.



Gambar 1. Algoritma RC4

E. Wireshark

Wireshark adalah penganalisa paket jaringan. Sebuah program analisa paket jaringan yang akan mencoba untuk menangkap paket jaringan dan mencoba untuk menampilkan data paket sedetail mungkin. Di masa lalu, *tool* semacam ini sangat mahal, eksklusif, atau keduanya. Wireshark adalah mungkin salah satu analisa terbaik paket open source yang tersedia saat ini.

III. Sekilas Kriptografi Skype

A. Registrasi

Kriptosistem Skype dimulai dengan registrasi pengguna. Pengguna memilih *username* yang diinginkan, A, dan password, P_A. Klien pengguna membangkitkan sepasang kunci RSA yaitu kunci privat, S_A dan kunci publik, V_A. Kunci privat, S_A, dan hash dari password, $H(P_A)$, disimpan seaman mungkin pada platform

pengguna.

Klien selanjutnya menetapkan sebuah *256-bit AES-encrypted session* dengan Server Pusat. Kunci untuk sesi ini dipilih oleh klien dengan bantuan dari platform-spesifik pembangkit bilangan acak. Klien dapat memverifikasi bahwa klien benar-benar berbicara dengan server. Klien mengirimkan server, antara lain, A, $H(P_A)$ dan V_A.

Server Pusat memutuskan apakah A adalah unik dan apakah dapat diterima berdasarkan aturan penamaan Skype. Jika diterima, server akan menyimpan (A, $H(H(P_A))$) dalam database. Server Pusat juga membentuk *Identity Certificate* untuk A, IC_A, yang berisi, antara lain, RSA Server Pusat *signature binding* A dan V_A, $\{A, V_A\}^{S}{}_{S}$ dan identifier kunci S_S. IC_A dikembalikan ke A.

B. Peer-to-Peer Key Agreement

Misalkan seorang pemanggil, A, ingin berkomunikasi dengan penerima, B, dan belum pernah ada sesi Skype sebelumnya antara mereka. Dalam hal ini sesi baru didirikan dan disediakan dengan kunci 256-bit sesi sendiri, SK_{AB}. Sesi ini akan ada selama ada lalu lintas di kedua arah antara A dan B, dan untuk beberapa waktu yang tetap sesudahnya. Setelah sesi selesai, SK tersebut masih dipertahankan dalam memori sampai klien ditutup.

Pembentukan sesi pertama harus menciptakan konektivitas antara A dan B di Skype *cloud*. Dengan menggunakan konektivitas ini, A dan B sekarang terlibat dalam sebuah *key agreement protocol* di mana, antara lain, mereka memeriksa perbaruan, memverifikasi identitas masing-masing, dan menyepakati SK_{AB}.

C. Kriptografi Sesi

Semua lalu lintas dalam sebuah sesi dienkripsi dengan meng-XOR-kan plainteks dengan *key stream* yang dihasilkan oleh 256-bit AES yang berjalan dalam *integer counter mode* (ICM). Kunci yang digunakan adalah SK_{AB}. Sesi Skype mengandung *multiple streams*.

IV. SERANGAN DALAM SKYPE *KEY AGREEMENT PROTOCOL*

Salah satu cara untuk menguji kekuatan *key agreement protocol* adalah dengan menyelidiki kemungkinan dari berbagai serangan terhadapnya.

A. Man-in-the-Middle (MITM) Attack

Tujuan dalam serangan ini adalah penyerang perantara, MITM, untuk meniru pemanggil dan/atau penerima satu sama lain. Kemudian, informasi akan diteruskan dari pemanggil ke penyerang ke penerima dan sebaliknya. Tujuan dari serangan ini adalah akses ke komunikasi seluruh pemanggil dan penerima yang tidak diketahui oleh pemanggil dan penerima bahwa penyadapan itu terjadi.

Dalam rangka untuk melakukan serangan MITM, penyerang harus mampu meyakinkan pemanggil bahwa dia adalah penerima (dan sebaliknya). Penyerang bisa melakukan ini dengan sebuah *signed certificate* valid yang menunjukkan *username* adalah callee (resp. pemanggil). Sertifikat ini dapat menjadi sertifikat nyata yang digunakan oleh penerima, atau sertifikat palsu. Penyerang juga harus mampu mencegat dan/atau memblokir semua lalu lintas antara pemanggil dan penerima.

Ada beberapa skenario serangan:

- Satu skenario mencegah sebuah sesi dari yang ditetapkan, tetapi tidak membahayakan kerahasiaan komunikasi.
- Dua skenario lainnya memerlukan baik penaklukan fisik, perangkat keras, dan mekanisme keamanan perangkat lunak pada *peer* yang berpartisipasi atau sebuah pra-komputasi yang tidak mungkin. Dengan persiapan tersebut, kedua skenario itu membutuhkan beberapa pencegatan yang diikuti oleh kedua pasca-komputasi yang tidak mungkin. Jika semua itu bisa dilakukan, penyerang dapat membahayakan keamanan sesi *peer-to-peer* tunggal.
- Skenario lain membutuhkan penaklukan keamanan di kedua *peers*. Dalam hal ini, semua sesi antara yang sepasang *peers* dapat dibahayakan.
- Skenario terakhir membutuhkan penaklukan mekanisme keamanan pada Server Pusat Skype.

B. Replay Attack

Sebuah *replay attack* berusaha untuk meyakinkan node untuk masuk ke sesi bersama penyerang dengan memutar data yang ditangkap oleh penyerang dari sesi sebelumnya antara target dan node lain. *Replay attack* kemungkinan bertujuan untuk menduplikasi *key stream* yang digunakan sebelumnya (yang dapat memungkinkan dilakukannya *cryptanalysis-at-depth*) dan memblokir sebuah node dari komunikasi dengan klien tertentu lainnya.

Penyerang bisa mengamati *multiple handshakes* yang melibatkan sebuah node target. Hal ini akan memberikan akses ke berbagai ajakan dan tanggapan. Penyerang kemudian bisa mengirim ajakan untuk target berpura-pura menjadi *peer* sebelumnya. Targetnya akan merespon dengan ajakan tersendiri. Jika ajakan target adalah identik dengan salah satu yang penyerang telah amati sebelumnya untuk pemanggil ini, penyerang kemudian bisa menjawab ajakan dengan benar dan melanjutkan ke aspek selanjutnya dari *key exchange protocol*. Namun, karena ajakan adalah sepanjang 64 bit dan dipilih secara acak, kemungkinan hal ini terjadi adalah rendah. Kesempatan mendapatkan ajakan ulang dari klien, dalam kasus beberapa pengamatan, jumlah pengamatan N terhadap total kemungkinan adalah N / 264.

Bahkan jika kejadian yang tidak mungkin ini terjadi, penyerang masih tidak memiliki akses ke kunci AES bahkan jika kejadian yang lebih tidak mungkin terjadi bahwa target memilih secara acak 128 bit kunci kontribusi yang sama karena memilih selama sesi yang direkam oleh

penyerang. Hal ini mungkin terjadi sekali setiap 2^{128} kali percobaan, probabilitasnya sangat rendah.

C. Password Guessing Attack

Pengguna bisa memilih apakah akan "mengingat" *password* Skype mereka pada platform yang mereka gunakan dan sebagian besar pengguna memilih opsi ini. Pada platform Windows, *password* diberikan kepada sistem operasi agar terlindungi di bawah Windows CryptProtectData API. Seorang pengguna yang kemudian dapat login ke Windows dapat langsung menggunakan Skype. Minoritas pengguna yang memilih untuk tidak mengingat password mereka pada komputer yang mereka gunakan harus login melalui protokol *client-server* sebelum mereka dapat menggunakan Skype. Untuk melindungi dari *password guessing attack*, Server Pusat Skype melakukan timeout setelah serangkaian sandi yang salah dimasukkan.

D. Kelemahan dalam Penggunaan CRC

Checksum jenis CRC (*Cyclic redundancy check*) yang umum digunakan dalam protokol komunikasi untuk mendeteksi bit error secara andal dan efisien. Namun, karena CRC linear, CRC mungkin tidak cocok untuk tujuan mendeteksi modifikasi data yang disengaja. Hal ini adalah salah satu masalah yang ditemukan di WEP, protokol keamanan asli untuk IEEE 802.11 wireless LAN. Beberapa aspek dari Skype menggunakan checksum jenis CRC Skype dengan cara yang mirip dengan WEP dan akibatnya dengan beberapa kelemahan yang sama.

E. Side-Channel Attacks

Telah diketahui bahwa implementasi dari operasi kriptografi kadang-kadang dapat membocorkan informasi tentang plainteks atau kunci melalui konsumsi *shared resources*, seperti *storage*, waktu atau *power* CPU. Klien Skype tidak membuat pertahanan terhadap serangan semacam ini. Jadi, misalnya, jika sebuah program berbahaya yang berjalan pada platform yang sama sebagai klien Skype, program berbahaya ini mungkin dapat menyimpulkan bit dari kunci privat pengguna. Hal ini pada akhirnya akan memungkinkan pemilik program jahat untuk menyamar sebagai pengguna.

F. ASN1 Attack

Beberapa tahun yang lalu, sekelompok peneliti Finlandia di Oulu University menemukan berbagai kerentanan berpotensi bahaya dalam agen SNMP (Simple Network Management Protocol) dari sejumlah produk vendor terkemuka. Sumber yang paling umum dari masalah ini adalah ketidakmampuan produk ini untuk melakukan parsing muatan encoded ASN1 (Abstract Syntax Notation One) secara aman dan benar. Tidak mengherankan jika banyak yang beralih ke penggunaan sertifikat SSL dari X509, sebagaimana protokol lain mengandalkan pada beberapa cara skema encoding tersebut.

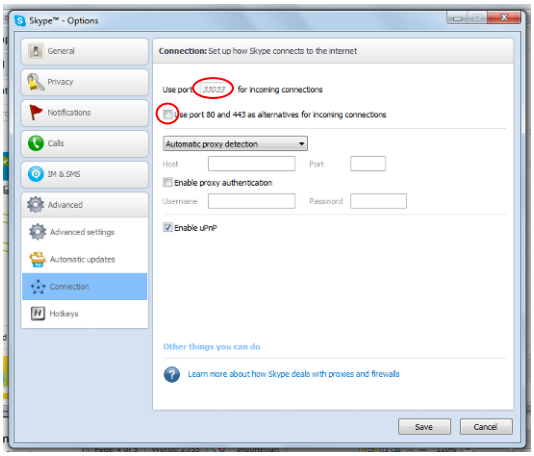
Protokol Skype tidak menggunakan ASN1, tetapi mereka menggunakan mekanisme yang serupa dan sangat bergantung pada kemampuan mereka untuk melakukan parsing muatan encoded dengan benar. Termasuk dalam muatan ini di mana penyerang dapat mengatur untuk hampir setiap nilai apapun. Telah ditemukan kesalahan potensial pada Skype yang terkait dengan decoding dari bilangan bulat. Kesalahan tidak membahayakan kerahasiaan komunikasi Skype, tapi mungkin menyebabkan perilaku tak terduga terhadap input berbahaya.

V. ANALISIS

A. Mengambil Paket Jaringan Skype

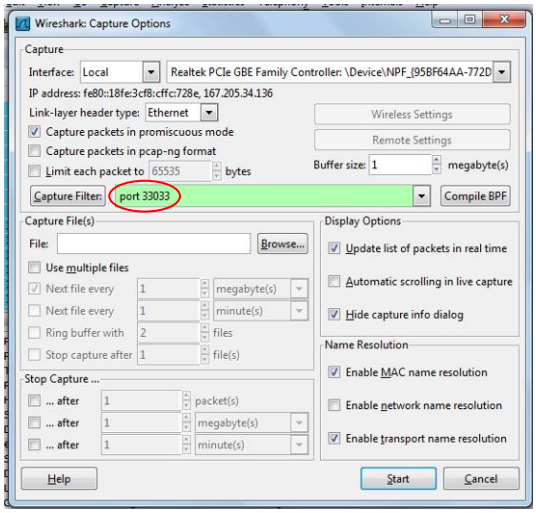
Data stream Skype dapat diperoleh dengan menggunakan *tool* Wireshark. Data stream Skype terbagi-bagi menjadi paket-paket jaringan yang dapat dipantau oleh Wireshark. Berikut ini langkah-langkah yang dilakukan untuk dapat mengambil paket-paket jaringan Skype:

1. Mengatur koneksi yang digunakan Skype agar dapat dipantau oleh Wireshark. Berdasarkan percobaan, Skype harus menggunakan *port* 33033 saja agar dapat dipantau oleh Wireshark. Cara mengubah *port* tersebut adalah pada aplikasi Skype pilih "*Tools*", kemudian pilih "*Options*", kemudian pilih "*Advanced: Connection*". Isi *port* 33033 pada pilihan "*Use port for incoming connection*" dan kosongkan centang pada pilihan "*Use port 80 and 443 as alternatives for incoming connections*".



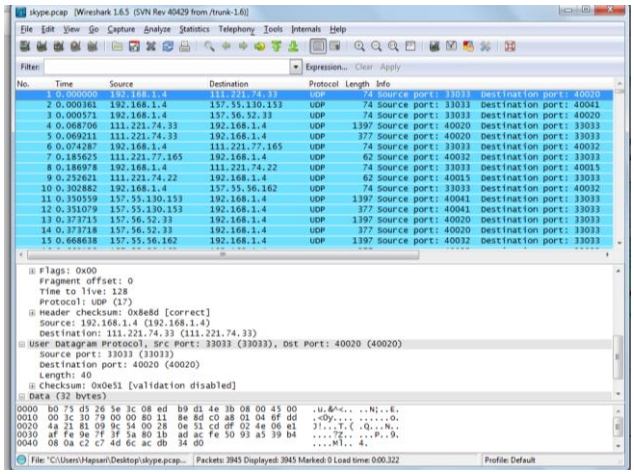
Gambar 2. Mengatur koneksi Skype

2. Mengatur pantauan Wireshark sesuai dengan port yang digunakan Skype. Pada Wireshark pilih "*Capture*", kemudian pilih "*Options*". Isi "Capture Filter" dengan "port 33033", pilih "Start".



Gambar 3. Mengatur *Capture Filter* Wireshark

3. Lakukan aktivitas panggilan pada Skype, misalkan lakukan panggilan pada *Echo / Sound Test Service*. Pada Wireshark akan muncul daftar paket-paket yang berhasil ditangkap.



Gambar 4. Daftar paket yang berhasil ditangkap Wireshark

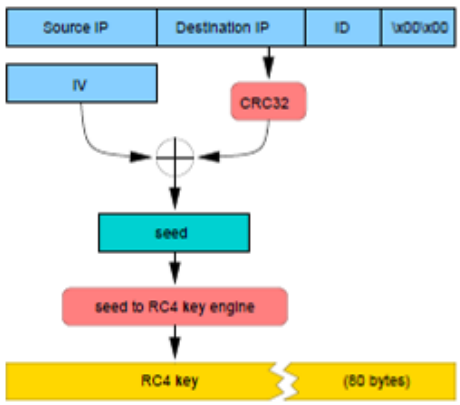
B. Analisis Paket Jaringan Skype

Berikut ini detail dari sebuah paket (paket pertama dari daftar):

```
  Frame 1: 74 bytes on wire (592 bits), 74 bytes
captured (592 bits)
      Arrival      Time:      Mar      26,      2013
12:12:11.778982000 SE Asia Standard Time
      Epoch Time: 1364274731.778982000 seconds
      [Time delta from previous captured frame:
0.000000000 seconds]
      [Time delta from previous displayed frame:
0.000000000 seconds]
      [Time since reference or first frame:
0.000000000 seconds]
      Frame Number: 1
      Frame Length: 74 bytes (592 bits)
      Capture Length: 74 bytes (592 bits)
      [Frame is marked: False]
      [Frame is ignored: False]
      [Protocols in frame: eth:ip:udp:data]
      [Coloring Rule Name: UDP]
      [Coloring Rule String: udp]
  Ethernet      II,      Src:      08:ed:b9:d1:4e:3b
(08:ed:b9:d1:4e:3b),      Dst:      Zte_26:5e:3c
(b0:75:d5:26:5e:3c)
      Destination:                        Zte_26:5e:3c
(b0:75:d5:26:5e:3c)
          Address:                        Zte_26:5e:3c
(b0:75:d5:26:5e:3c)
          .... ...0 .... .... .... .... = IG bit:
Individual address (unicast)
          .... ..0. .... .... .... .... = LG bit:
Globally unique address (factory default)
      Source:                        08:ed:b9:d1:4e:3b
(08:ed:b9:d1:4e:3b)
          Address:                08:ed:b9:d1:4e:3b
(08:ed:b9:d1:4e:3b)
          .... ...0 .... .... .... .... = IG bit:
Individual address (unicast)
          .... ..0. .... .... .... .... = LG bit:
Globally unique address (factory default)
      Type: IP (0x0800)
  Internet Protocol Version 4, Src: 192.168.1.4
(192.168.1.4), Dst: 111.221.74.33 (111.221.74.33)
      Version: 4
      Header length: 20 bytes
      Differentiated Services Field: 0x00 (DSCP
0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable
Transport))
      Total Length: 60
      Identification: 0x3079 (12409)
      Flags: 0x00
      Fragment offset: 0
      Time to live: 128
      Protocol: UDP (17)
      Header checksum: 0x8e8d [correct]
      Source: 192.168.1.4 (192.168.1.4)
      Destination: 111.221.74.33 (111.221.74.33)
  User Datagram Protocol, Src Port: 33033 (33033),
Dst Port: 40020 (40020)
      Source port: 33033 (33033)
      Destination port: 40020 (40020)
      Length: 40
      Checksum: 0x0e51 [validation disabled]
  Data (32 bytes)

  0000  cd df 02 4e 06 e1 af fe 9e 7f 3f 5a 80 1b
ad ac   ...N......?Z....
  0010  fe 50 93 a5 39 b4 08 0a c2 c7 4d 6c ac db
34 d0   .P..9.....Ml..4.
      Data:
cddf024e06e1affe9e7f3f5a801badacfe5093a539b4080a...
    [Length: 32]
```

Data dari setiap paket (bagian yang diberi warna merah) dienkripsi dengan menggunakan algoritma RC4. Kunci RC4 dibangkitkan dengan memperhitungkan elemen-elemen dari datagram: IP sumber dan IP tujuan, ID paket Skype, serta *obfuscation layer's IV* Skype.

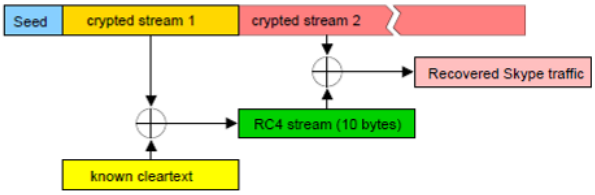


Gambar 5. Proses pembangkitan kunci RC4

Kunci RC4 adalah sepanjang 80 byte, namun diperkirakan ada paling banyak 2^{32} kemungkinan kunci. Jumlah kemungkinan kunci ini sangat banyak sehingga tidak mungkin untuk dipecahkan. Sampai saat ini tidak ada yang dapat memecahkan RC4 sehinggat dapat dikatakan sangat kuat.

Namun RC4 dengan mudah dapat diserang dengan *known-plaintext attack*, dengan cara meng-XOR-kan dua set byte cipherteks. *Known-plaintext attack* kemungkinan dapat dilakukan dengan adanya fakta-fakta berikut:

- Hampir semua lalu lintas jaringan Skype dienkripsi, namun tidak semuanya.
- Komunikasi UDP menyiratkan lalu lintas yang jelas untuk mempelajari IP publik.
- Komunikasi TCP menggunakan aliran RC4 yang sama dua kali.
- *Stream* TCP dimulai dengan muatan yang panjangnya 14 byte.
- Dari sini dapat ditemukan 10 byte dari stream RC4.
- Stream RC4 digunakan dua kali dan telah diketahui 10 dari 14 byte pertama.



Gambar 5. *Known-plaintext attack* pada RC4

VI. KESIMPULAN

Para desainer dari Skype tidak ragu-ragu untuk menggunakan kriptografi secara luas dan baik dalam rangka membangun dasar kepercayaan, keaslian, dan kerahasiaan untuk layanan *peer-to-peer* mereka. Para pengembang Skype menerapkan fungsi kriptografi dengan benar dan efisien. Akibatnya, kerahasiaan sesi Skype jauh lebih besar daripada yang ditawarkan oleh panggilan telepon kabel atau nirkabel atau melalui email dan lampiran email.

Namun banyak yang mengatakan bahwa proses dalam Skype seperti *blackbox* karena kurangnya transparansi. Sampai saat ini tidak ada cara untuk mengetahui apakah ada atau akan ada *backdoor*.

Dalam analisis telah dicoba bahwa bukan tidak mungkin untuk mengambil paket jaringan Skype. Namun paket jaringan ini tidak memberikan informasi apapun karena berupa cipherteks, dienkripsi dengan algoritma RC4. Walaupun algoritma RC4 adalah algoritma yang sangat kuat, bukan tidak mungkin untuk dilakukannya kriptanalis karena algoritma RC4 rentan terhadap *known-plaintext attack*.

Dari hasil analisis dapat disimpulkan bahwa sistem keamanan Skype dapat dipercaya karena sudah menggunakan kriptografi dengan baik, terlepas dari kemungkinan adanya atau akan adanya *backdoor*.

REFERENCES

[1] Biondi, P. & Desclaux, F. (2006). *Silver Needle in the Skype*. http://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-biondi/bh-eu-06-biondi-up.pdf, diakses tanggal 25 Maret 2013.

[2] http://www.skype.com/en/security/#encryption, diakses tanggal 25 Maret 2013.

[3] Berson, T. (2005). *Skype Security Evaluation*. http://download.skype.com/share/security/2005-031%20security%20evaluation.pdf, diakses tanggal 25 Maret 2013.

[4] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, diakses tanggal 25 Maret 2013.

[5] http://blog.uin-malang.ac.id/goji/files/2011/03/goji-wireshark.pdf, diakses tanggal 25 Maret 2013.

[6] Munir, Rinaldi. (2005). *Kriptografi*. Bandung : Penerbit ITB.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 Maret 2013

Hapsari Tilawah 13509027