Implementasi Algoritma Kriptografi Kunci-Publik RSA sebagai *Plug-in* Aplikasi *Instant messaging* Yahoo Messenger

David Susanto [1)]

1) Jurusan Teknik Informatika ITB, Bandung 40132, email: if14019@students.if.itb.ac.id

Abstraksi – *Instant messaging merupakan salah satu layanan yang sangat populer di kalangan pengguna internet saat ini. Sebagai alat komunikasi di internet, instant messaging relatif sederhana dan mudah dalam penggunaannya. Namun di balik kesederhanaan dan kemudahan dalam penggunaannya, instant messaging memiliki kelemahan dalam keamanan data yang dipertukarkan. Instant messaging sangat rawan terhadap penyadapan (eavesdropping) data. Hingga saat ini, kelemahan tersebut masih belum begitu disadari oleh pengguna, padahal penggunaan instant messaging saat ini sangat luas, mulai dari obrolan ringan hingga percakapan penting yang bernilai bisnis tinggi. Pada makalah ini akan dibahas bagaimana mengimplementasikan algoritma kriptografi kunci publik RSA untuk mengamankan percakapan antara dua pihak pada aplikasi instant messaging Yahoo Messenger. Implementasi yang dilakukan berupa plug-in untuk aplikasi tersebut. Plug-in tersebut bekerja dengan mengenkripsi setiap pesan memanfaatkan kunci publik penerima sebelum dikirimkan ke server penyedia layanan instant messaging. Kemudian saat pesan telah sampai di pihak penerima, pesan tersebut akan didekripsi kembali dengan kunci privat si penerima sesuai algoritma RSA. Dengan demikian diharapkan penyadap pesan tidak akan mendapatkan informasi yang mereka inginkan dari sebuah percakapan instant messaging karena pesan yang dikirimkan berupa cipherteks.*

.

Kata Kunci*: eavesdropping, penyadapan, instant messaging, Yahoo Messenger, RSA, Yahoo Messenger Plug-in SDK.*

1. PENDAHULUAN

Internet sebagai teknologi yang dekat dengan kehidupan masyarakat modern telah mengubah perilaku manusia dalam berkomunikasi dan mengakses informasi. Jika pada zaman dahulu manusia mengenal media komunikasi seperti surat pos dan telegraf, maka saat ini telah tersedia banyak alternatif media komunikasi melalui internet. Sebut saja misalnya surat elektronik atau yang lebih populer dikenal dengan nama email. Email sebenarnya telah ada sejak lahirnya internet. Namun, email mulai populer dipergunakan sejak era 1980-an hingga saat ini. Walaupun demikian, email memiliki karakteristik dan keterbatasan sebagai media komunikasi yang sifatnya tidak interaktif. Pengirim email tidak dapat serta-merta memperoleh balasan atas email yang ia kirimkan. Selain itu percakapan juga akan terasa merepotkan, karena setiap kali akan mengirim pesan, pengguna harus memasukkan alamat email penerima pesan atau menekan tombol balas (*reply*) kemudia memasukkan pesan yang akan dikirimkan lalu baru menekan tombol kirim (*sent*) pada aplikasi email client miliknya agar email dapat dikirimkan. Hal ini tentu saja sangat mengurangi interaktifitas apabila pengguna ingin melakukan percakapan (*chat*) dengan berbalas pesan.

Hal tersebut kemudian diatasi dengan diperkenalkannya *instant messaging* sebagai media komunikasi digital lain yang memanfaatkan teknologi internet. *Instant messaging* mempergunakan pendekatan percakapan daripada pendekatan surat-menyurat seperti pada email. Pengguna *instant messaging* dapat bertukar pesan satu sama lain secara cepat sehingga lebih interaktif dan menarik untuk melakukan percakapan. Oleh karena itu sangatlah wajar jika *instant messaging* dapat dengan cepat populer di kalangan pengguna internet.

Belum dapat dipastikan dengan tepat kapan teknologi *instant messaging* lahir. Ada sumber yang menyebutkan bahwa *instant messaging* sebenarnya telah lahir sebelum internet ada. *Instant messaging* 'primitif' tersebut lahir pada pertengahan 1960-an dan beroperasi pada sistem operasi CTTS dan Multics. Pada awalnya, CTTS memiliki sistem yang berfungsi sebagai notifikasi kepada pengguna untuk layanan *printing*. Namun, fungsi tersebut dengan cepat beralih fungsi menjadi media komunikasi diantara pengguna yang *login* pada mesin yang sama. Seiring dengan perkembangan teknologi jaringan komputer, bermunculan protokol-protokol komunikasi *instant messaging* lainnya. Ada protokol yang sifatnya *peer-to-peer* seperti *talk*, *ntalk*, dan *ytalk*. Ada juga protokol yang mengharuskan penggunanya untuk terkoneksi pada sebuah *server* seperti *talker* dan *IRC*.

Namun demikian, *instant messaging* baru mulai populer saat internet mulai luas dipergunakan pada pertengahan dekade 1990-an. *Instant messaging* mulai benar-benar meledak di dunia internet pada bulan November 1996 saat Mirabilis memperkenalkan ICQ, sebuah aplikasi *instant messaging* yang sifatnya gratis dan dapat dipergunakan siapa saja. ICQ, yang merupakan singkatan pengucapan "*I Seek You*",

memiliki tampilan grafis untuk antarmuka pengguna (GUI) sehingga dapat dengan mudah diterima secara luas oleh pengguna internet saat itu.

Pada tahun 1997, AOL sebagai salah satu pionir dalam komunitas *online* yang berpusat di Amerika Serikat meluncurkan AIM yang merupakan singkatan dari AOL *Instant messaging*. AIM menawarkan layanan tambahan seperti percakapan dengan beberapa pengguna secara bersamaan dalam sebuah chat room. Kemudian pada bulan Juni 1998, AOL mengakuisisi Mirabilis dan ICQ-nya. Model ICQ inilah yang kemudian menjadi dasar berkembangnya banyak aplikasi *instant messaging* lainnya yang beredar saat ini seperti Yahoo Messenger, Windows Live Messenger dan Google Talk.

Pada umumnya, aplikasi *instant messaging* memiliki fitur-fitur yang relatif sama dengan aplikasi *instant messaging* yang lain. Berikut adalah fitur-fitur yang umumnya terdapat dalam sebuah aplikasi *instant messaging*:

1. Pesan instan, pengguna dapat bertukar pesan secara cepat dan interaktif.
2. Daftar teman, pengguna dapat menyimpan daftar teman-temannya dan dapat melihat status temannya tersebut apakah sedang *online* atau tidak.
3. Konferensi, beberapa pengguna dapat melakukan percakapan secara besama-sama di dalam sebuah *chat room*.
4. Transfer file, pengguna dapat mengirimkan file kepada pengguna lainnya.
5. *Video*/*voice call*, pengguna dapat melakukan percakapan secara langsung dengan suara dan tampilan gambar bergerak dari lawan bicaranya.

Saat ini penggunaan *instant messaging* sudah sangat luas digunakan. Anak-anak, remaja, hingga orang dewasa mempergunakan teknologi ini. Tujuan penggunaannya juga bermacam-macam. Ada yang hanya sekadar untuk bercakap-cakap dengan teman dan kolega dan ada juga yang dipergunkan untuk kepentingan yang lebih serius seperti konferensi online sesama rekan kerja dalam sebuah perusahaan. Bahkan belakangan ini banyak perusahaan yang menyediakan jalur *instant messaging* sebagai layanan penggunanya (*customer service*) disamping mempergunakan telepon dan email.

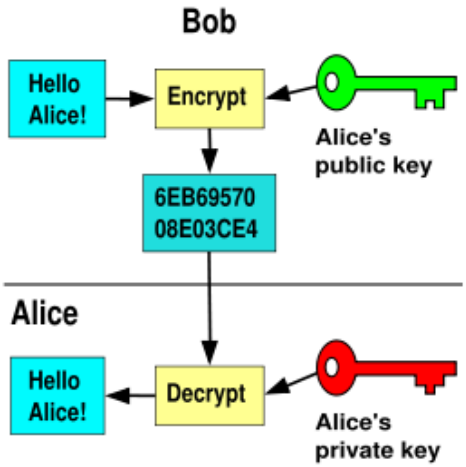
Walaupun penggunaan *instant messaging* relatif mudah dan menyenangkan, namun celah keamanan yang dimiliki oleh *instant messaging* cukup banyak. Salah satu yang paling sering terjadi ada penyadapan percakapan (*eavesdropping*). Sang penyadap dapat dengan mudah mengetahui isi pembicaraan dalam *instant messaging* yang tentunya sangat merugikan pengguna. Apalagi jika pengguna *instant messaging* kurang hati-hati dan membocorkan informasi berharganya seperti password dan nomor kartu kredit melalui *instant messaging*. Pembicaraan yang bernilai bisnis tinggi dan sifatnya rahasia yang berlangsung melalui *instant messaging* tentunya sangat tidak aman untuk dilakukan. Untuk itu, pada makalah ini akan dibahas mengenai implementasi algoritma RSA untuk mencegah penyadapan percakapan pada *instant messaging* Yahoo Messenger. Yahoo Messenger dipilih karena saat ini merupakan aplikasi *instant messaging* yang paling populer (dipergunakan oleh lebih dari 60 juta pengguna) dan memiliki fasilitas plug-in yang dapat dibuat oleh pihak ketiga.

2. DASAR TEORI

2.1. Kriptografi Kunci-Publik

Sebelum ditemukannya algoritma kriptografi kunci-publik pada tahun 1975, hanya dikenal algritma kriptografi kunci-simetri. Pada algoritma kriptografi kunci-simetri kunci yang dipergunakan kunci yang sama untuk mengenkripsi dan mendekripsi pesan. Oleh karena itu, kerahasiaan kunci ini harus dijaga oleh kedua belah pihak (pengirim dan penerima pesan). Hal ini menyebabkan masalah dalam distribusi kunci dari pengirim ke penerima pesan. Bagaimana caranya pengirim pesan dapat memberitahukan kunci yang ia pergunakan kepada penerima pesan. Jika menggunakan saluran publik (misalnya internet, telepon atau pos) tentunya sangat tidak aman karena rawan penyadapan, sehingga diperlukan kurir atau bertemu secara langsung untuk membagikan kunci tersebut. Cara ini tentunya mahal dan lambat.

Kemudian Diffie dan Hellman mengusulkan sistem kriptografi asimetri (*asymmetric cryptosystem*) yang memungkinkan pengguna dapat berkomunikasi secara aman tanpa perlu berbagi kunci rahasia. Sistem kriptografi ini juga dikenal dengan nama kriptografi kunci-publik (*public-key cyptography*).



Gambar 1: Enkripsi/dekripsi kriptografi kunci-publik

Kriptografi kunci-publik memanfaatkan prinsip yang sederhana namun elegan. Kriptografi ini memanfaatkan pasangan kunci, yaitu kunci-publik dan kunci-privat. Kunci-publik, *e*, sifatnya tidak rahasia dan dapat diberitahukan kepada siapa saja, sementara kunci-privat, *d*, sifatnya rahasia dan hanya boleh diketahui oleh pemiliknya saja. Karena kunci-publik tidak sama dengan kunci-privat, maka kriptografi ini dinamakan kriptografi asimetri.

Untuk mengirimkan pesan rahasia kepada penerima pesan, pengirim pesan harus mengenkripsi pesan tersebut dengan mempergunakan kunci-publik si penerima. Kemudian pesan yang didapatkan oleh penerima pesan dapat didekripsi dengan memanfaatkan kunci-privat miliknya. Sehingga dalam hal ini pengirim dan penerima pesan tidak melakukan pertukuran kunci rahasia, melainkan hanya memberitahukan kunci-publik satu sama lain untuk melakukan komunikasi. Apabila *E* merupakan fungsi enkripsi dan *D* merupakan fungsi dekripsi, maka untuk melakukan enkripsi plainteks *m* dan dekripsi cipherteks *c* dapat dirumuskan dengan $E_d(m) = c$ dan $D_d(c) = m$.

2.2. Algoritma Kriptografi Kunci-Publik RSA

RSA merupakan salah satu algoritma kunci-publik yang paling populer diimplementasikan pada berbagai aplikasi. Algoritma RSA diciptakan pada tahun 1976 oleh tiga orang peneliti dari *Massachussets Insititute of Technology* (MIT), yaitu *Ron Rivest*, *Adi Shamir* dan *Leonard Adleman*. Nama RSA diambil dari gabungan inisial nama belakang ketiga penemunya tersebut. Sulitnya untuk memfaktorkan bilangan besar menjadi faktor-faktor prima dimanfaatkan oleh algoritma RSA untuk menjamin keamanan algoritma kriptografi ini.

2.2.1. Pembangkitan Pasangan Kunci

RSA melibatkan kunci-publik dan kunci-privat dalam proses enkripsi dan dekripsinya. Kunci-publik sifatnya tidak rahasia dan boleh diketahui oleh orang lain dan dipergunakan untuk mengenkripsi pesan. Pesan yang dienkripsi dengan kunci-publik dapat didekripsi dengan mempergunakan kunci-privat. Pembangkitan kunci untuk algoritma RSA dilakukan dengan cara berikut:

1. Pilih dua buah bilangan prima sembarang, *p* dan *q*.
2. Hitung $n = p.q$ (sebaiknya $p \neq q$, sebab jika $p = q$ maka $n = p^2$ sehingga *p* dapat diperoleh dengan menarik akar pangkat dua dari *n*).
3. Hitung $\phi(n) = (p - 1)(q - 1)$.
4. Pilih kunci public, *e*, yang relatif prima terhadap $\phi(n)$.
5. Bangkitkan kunci privat dengan menggunakan persamaan $e.\ d \equiv 1 \pmod{\phi(n)}$.

Algoritma di atas akan menghasilkan pasangan kunci yaitu:

1. Kunci-publik adalah pasangan (*e*, *n*).
2. Kunci-privat adalah pasangan (*d*, *n*).

2.2.2. Algoritma Enkripsi/Dekripsi Pesan

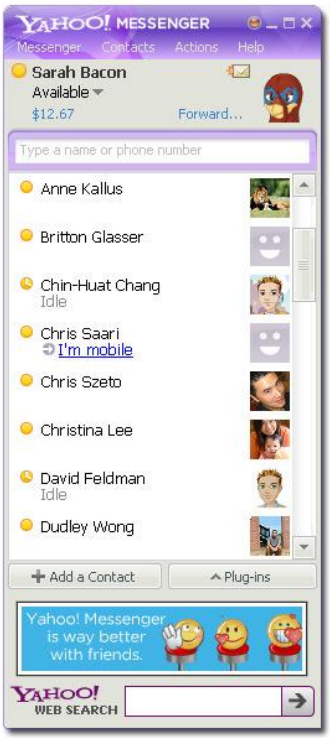
Berikut adalah algoritma enkripsi pesan yang dilakukan dalam algoritma kriptografi RSA:

1. Ambil kunci-publik penerima pesan, yaitu e dan modulus n.
2. Nyatakan plainteks *m* menjadi blok-blok m_1, m_2, …, m_n sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [*0*, *n-1*].
3. Setiap blok m, dienkripsi menjadi blok c_i dengan rumus $c_i = m_i^e \bmod n$.

Sementara untuk melakukan dekripsi terhadap cipherteks cukup dengan memanfaatkan rumus $m_i = c_i^d \bmod n$ pada setiap blok c_i untuk didekripsi menjadi blok m_i.

3. YAHOO MESSENGER

Yahoo Messenger merupakan layanan *instant messaging* yang disediakan oleh Yahoo, sebuah perusahaan teknologi informasi yang memiliki layanan utama portal internet yang terbesar saat ini. Yahoo Messenger saat ini dipergunakan oleh lebih dari 60 juta pengguna yang kabarnya merupakan layanan *instant messaging* terbesar pada saat tulisan ini dibuat.



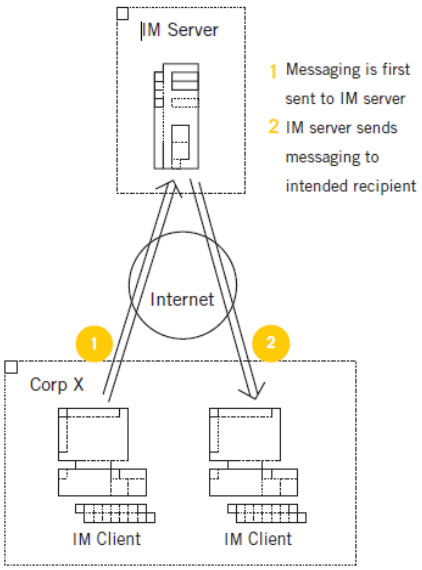
Gambar 2: Tampilan aplikasi Yahoo Messenger client

Pengguna Yahoo Messenger diidentifikasi dengan menggunakan Yahoo ID, yaitu *Single Sign-On* (SSO) account yang digunakan untuk mengakses layanan Yahoo yang lain seperti Yahoo Webmail, Yahoo Groups dan Yahoo 360. Yahoo Messenger dilengkapi dengan fasilitas *Voice Chat* dan Webcam, sehingga pengguna dapat berkomunikasi dengan suara dan video. Selain itu, pengguna Yahoo dapat mengalihkan pesan yang diterimanya ke ponsel melalui SMS (*Short Message Service*). Salah satu fitur yang khas pada Yahoo Messenger adalah IMvironments yang memungkinkan pengguna untuk mengubah tampilan jendela percakapan. Yahoo Messenger pun memiliki fitur avatar, sehingga pengguna dapat memilih gambar atau foto yang mewakili pengguna di dunia maya.

3.1. Arsitektur Sistem Yahoo Messenger

Arsitektur sistem yang digunakan oleh Yahoo Messenger adalah client-*server*. Arsitektur ini mewajibkan pengguna untuk login terlebih dahulu ke sebuah *server* yang memberikan layanan Yahoo Messenger dari aplikasi client yang terpasang pada komputer pengguna.

Segala pesan yang dikirimkan dari pengirim kepada lawan bicaranya selalu melalui *server* miliki Yahoo.

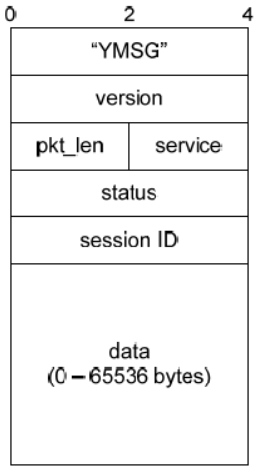


Gambar 3: Arsitektur sistem Yahoo Messenger

3.2. Protokol YMSG

YMSG merupakan protokol yang dipergunakan dalam sistem Yahoo Messenger. YMSG bukan merupakan protokol yang sifatnya opensource. Yahoo tidak pernah mempublikasikan dokumentasi teknis dari YMSG ini. Namun, beberapa orang melakukan rekayasa balik (*reverse engineering*) untuk menganalisa dan mempelajari protokol ini.

Protokol YMSG mengatur komunikasi antara aplikasi client dan *server* dengan mempergunakan koneksi TCP/IP melalui port 5050 sebagai port *default*. Apabila port ini diblok, maka akan dipergunakan port lainnya. Misalnya pada jaringan yang memiliki firewall, YMSG akan memanfaatkan HTTP untuk melakukan komunikasi.



Gambar 4: Struktur paket YMSG

Setelah melakukan login, client akan tetap tersambung dengan *server* selama koneksi TCP/IP tidak terputus. Demikian pula dengan client yang memanfaatkan HTTP untuk koneksinya, client akan tetap tersambung hingga client gagal mengirimkan request ping ke *server* (pesan ping dikirimkan setiap 30 detik).

Paket pesan pada YMSG terdiri dari header yang berukuran 20 byte diikuti dengan data yang ukurannya bervariabel dari 0 hingga 65.536 byte yang direpresentasikan dengan ASCII.

Beberapa bagian dari YMSG bergantung pada protokol lain. Misalnya untuk transfer file, diawali dengan protokol YMSG, kemudian untuk proses transfernya dilakukan melalui protokol HTTP.

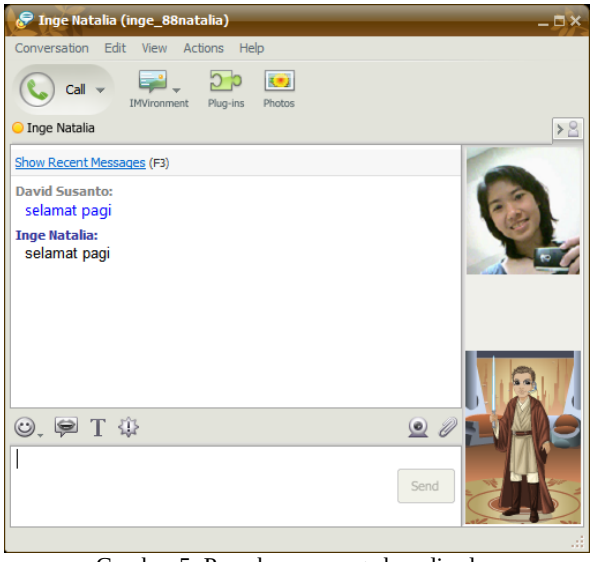
3.3. Analisis Keamanan Yahoo Messenger

Symantec Enterprise Security menyatakan beberapa serangan yang rentan pada layanan *instant messaging*, yaitu:

1. Penyadapan (*eavesdropping*), karena kebanyakan sistem *instant messaging* tidak mengenkripsi data yang dikirim, maka pihak ketiga bisa menyadap komunikasi yang berlangsung antara pengguna *instant messaging* dengan menggunakan packet *sniffer* atau teknologi sejenis.
2. Pembajakan *account* (*account hijacking*), kebanyakan sistem *instant messaging* rentan terhadap pembajakan account atau penyamaran (*spoofing*). Seseorang dapat membajak account orang lain atau menyamar menjadi orang tersebut. Proteksi terhadap password pada sistem *instant messaging* juga sangat lemah. Terkadang password pengguna disimpan dalam komputer klien, baik dalam bentuk terenkripsi atau tidak. Walaupun tersimpan dalam bentuk terenkripsi, password pengguna masih dapat di-*crack* dengan bantuan kakas yang sesuai.

3. Pengaksesan & pengubahan data pengguna. Sebagaimana perangkat lunak yang terhubung ke internet lainnya, program *instant messaging* dapat memiliki *bug* yang memungkinkan penyerangan melalui jaringan, seperti memanfaatkan *buffer overflow* atau paket data yang sudah dimodifikasi sedemikian rupa. Sehingga penyerang dapat mengambil alih komputer pengguna.
4. *Worm* dan serangan kombinasi .Seperti email, *instant messaging* memungkinkan penyebaran *worm* dan virus. Contoh kasus nyata adalah *worm* pada jaringan IRC.

Untuk makalah ini, akan ditekankan pada risiko keamanan penyadapan (*eavesdropping*). Penyadapan pesan percakapan sering sekali terjadi pada Yahoo Messenger. Hal ini terjadi karena YMSG tidak mendukung enkripsi pesan yang dikirimkan. Akibatnya pesan yang dikirimkan dapat dibaca dengan mudah oleh penyadap pesan menggunakan aplikasi *sniffer*, padahal pesan dikirimkan melalui jaringan publik pada saat menuju *server* dan pada saat dari *server* menuju penerima pesan.



Gambar 5: Percakapan yang akan disadap

Penulis melakukan percobaan untuk menyadap percakapan singkat melalui Yahoo Messenger dengan menggunakan aplikasi Wireshark. Aplikasi Wireshark ini akan menangkap paket data yang melewati jaringan kemudian membuka isinya. Pada percakapan gambar 5, penulis berhasil menyadap isi percakapan yang terjadi, seperti yang dapat dilihat pada gambar 6 dan gambar 7.

```
6f 6e 74 20 66 61 63 65   ..[31m<f ont face
55 49 22 3e 73 65 6c 61   ="Segoe  UI">sela
c0 80 59 4d 53 47 00 10   mat pagi ..YMSG..
00 16 88 47 0c 28 34 39   ...?.K..  ...G.(49
```

Gambar 6: Hasil penyadapan percakapan baris 1

```
54 69 72 65 30 74 45 65   Q6cjqaVS Tire0tEe
c0 80 31 c0 80 31 34 c0   b2gi..97 ..1..14.
20 70 61 67 69 c0 80 36   .selamat  pagi..6
34 c0 80 30 c0 80 00      3..;0..6 4..0...
```

Gambar 7: Hasil peyadapan percakapan baris 2

Terlihat dengan jelas, bahwa pesan yang dikirimkan melalui Yahoo Messenger dapat dengan mudah dibaca oleh penyadap pesan. Hal ini tentu saja sangat berbahaya apabila isi percakapan tersebut memiliki informasi yang sangat berharga.

Untuk mengatasi hal tersebut bisa dilakukan dua pendekatan sebagai solusi, yaitu:
1. Melewatkan pesan-pesan yang ada dalam percakapan Yahoo Messenger di dalam protokol HTTPS.
2. Mengenkripsi setiap pesan yang akan dikirimkan dan mendekripsi kembali pesan tersebut setelah diterima.

Pendekatan pertama susah untuk dilakukan, karena hingga saat ini YMSG tidak mendukung HTTPS untuk melewatkan pesan-pesan ke jaringan. Sehingga kita perlu melakukan perombakan protokol YMSG yang tentu saja hanya dapat dilakukan oleh Yahoo sendiri sebagai pemilik protokol YMSG.

Oleh karena itu, diambil pendekatan kedua, di mana setiap pesan akan dienkripsi terlebih dahulu sebelum dilewatkan pada jaringan. Kemudian pesan yang diterima akan didekripsi kembali agar dapat dibaca oleh penerimanya. Dengan demikian apabila terjadi penyadapan, pesan yang diperoleh si penyadap tidak akan memiliki arti, sehingga bocornya informasi dapat dicegah.

4. IMPLEMENTASI

4.1. Yahoo Messenger Plug-in SDK

Yahoo Messenger Plug-in SDK merupakan API dalam JavaScript dan C++ yang memungkinkan pihak ketiga untuk menciptakan *add-on* dengan fitur yang kolaboratif untuk dijalankan pada Yahoo Messenger. Plug-in yang dihasilkan dapat berjalan pada jendela utama Yahoo Messenger atau pada bagian percakapan. Pembuatan plug-in dengan menggunakan SDK ini dapat menghasilkan *add-on* sangat sederhana seperti halaman *web* HTML maupun *add-on* yang sangat rumit seperti peta interaktif yang memanfaatkan ActiveX, Flash, DHTML, AJAX dan teknologi berbasis *browser* lainnya.

Plug-in yang dihasilkan akan berupa sebuah file HTML dengan kode JavaScript yang memanggil API dari plug-in. Sebuah file teks *manifest* menspesifikasikan konfigurasi dan aturan untuk plug-in tersebut, misalnya di mana Yahoo Messenger dapat menemukan plug-in tersebut pada saat *runtime*.

Sehingga sebuah plug-in terdiri dari dua komponen, yaitu:

1. HTML, mengandung kombinasi tag HTML dan JavaScript implementasi *plug-in*.
2. *Manifest*, mengandung konfigurasi untuk masing-masing plug-in. Setiap *plug-in* memiliki sebuah file *manifest* masing-masing.

4.2. Implementasi Algoritma RSA sebagai Plug-in

Untuk implementasi algoritma RSA pada Yahoo Messenger, dipergunakan API yang terdapat pada Yahoo Messenger Plug-in SDK. Namun tidak semua fungsi dan objek dipergunakan untuk implementasi ini. Hanya beberapa objek dan fungsi relevan yang dipergunakan. Objek dan fungsi yang dipergunakan untuk implementasi ini adalah:

1. **IMMessage** dan **IIMessage.Text**, merupakan objek yang akan dipergunakan untuk mengakses isi string pesan percakapan yang akan dienkripsi dan didekripsi.
2. **_IEventsConversationWindow** (IMSent dan. Incoming IM), merupakan event handler yang dipergunakan untuk mengaktifkan fungsi enkripsi dan dekripsi pesan saat pesan dikirim dan diterima.
   a. ***Event* IMSent** akan dijalankan saat pesan dikirmkan kepada pengguna lain. Pada saat ini lah pesan akan dienkripsi sebelum dikirimkan.
   b. ***Event* IncomingIM** akan dijalankan ketika pesan diterima namun belum diproses untuk ditampilkan. Pada saat inilah pesan akan didekripsi kembali agar dapat dibaca oleh penerima pesan.

Untuk membangkitkan kunci, diimplementasikan aplikasi pembangkit pasangan kunci yang nantinya akan dipergunakan saat enkripsi dan dekripsi pesan. Sementara algoritma RSA diimplementasikan dengan menggunakan bahasa C++ sebagai objek ActiveX.

Penggunaan plug-in diawali dengan pengguna harus membangkitkan pasangan kunci-publik dan kunci-privat dengan aplikasi pembangkit pasangan kunci. Pasangan kunci yang dihasilkan akan disimpan menjadi file teks terpisah di direktori Yahoo Messenger pada komputer pengguna. Kemudian pengguna cukup menyalakan Yahoo Messenger dan mengirim pesan kepada pengguna lain yang juga menggunakan plug-in ini.

Sebelum mengirim pesan kepada lawan bicara akan dilakukan pertukaran kunci-publik terlebih dahulu. Kunci-publik lawan bicara akan disimpan sebagai file teks dalam komputer pengguna. Setelah itu pengguna dapat bertukar pesan. Pesan akan dienkripsi dengan kunci-publik lawan bicara sebelum dikirimkan dan pesan yang diterima akan didekripsi dengan kunci-privat penerima sebelum ditampilkan untuk dibaca.

```
0c 28 35 c0 80 73 75 70   .......G .(5..sup
67 61 c0 80 34 c0 80 69   erdavesa ga..4..i
74 61 6c 69 61 c0 80 32   nge_88na talia..2
33 37 c0 80 76 66 6b 69   06..2..3 37..vfki
33 65 41 39 67 70 72 76   2FowfBqN 3eA9gprv
33 71 5f 69 59 52 33 37   6db5CqML 3q_iYR37
54 69 72 65 30 74 45 65   Q6cjqaVS Tire0tEe
c0 80 31 c0 80 31 34 c0   b2qi..97 ..1..14.
```

Gambar 8: Prosedur enkripsi berbasis DNA

Setelah menggunakan plug-in ini, penulis mencoba menyadap percakapan antara penulis dan pengguna Yahoo Messenger lain yang menggunakan plug-in ini. Ternyata hasil penyadapan tidak mendapatkan informasi mengandung makna.

5. KESIMPULAN

1. Percakapan melalui Yahoo Messenger memiliki risiko keamanan yaitu sangat rentan terhapad penyadapan pesan (eavesdropping).
2. Protokol YMSG yang dipergunakan oleh Yahoo Messenger tidak melakukan enkripsi terhadap pesan dalam percakapan.
3. Pembuatan plug-in enkripsi dan dekripsi pesan dengan algoritma RSA merupakan salah satu solusi yang dapat diambil untuk mengatasi penyadapan pesan.
4. Plug-in yang dihasilkan memiliki keterbatasan di mana plug-in ini dapat berjalan hanya pada sesama pengguna dengan aplikasi client resmi dari Yahoo Messenger (bukan pihak ketiga) dan pada client penerima dan pengirim pesan telah dipasangkan plug-in enkripsi dan dekripsi pesan ini.

DAFTAR REFERENSI

[1] Munir, Rinaldi. *Diktat Kuliah IF5054 Kriptografi*. Institut Teknologi Bandung, 2006.

[2] Karhendana, Arie. *Keamanan pada Layanan Instant messaging: Studi Kasus Yahoo Messenger, Windows Live Messenger, dan Google Talk*. Institut Teknologi Bandung, 2006.

[3] *PC World - Security Flaw Found in Yahoo Messenger*. *http://www.pcworld.com.* Tanggal akses:12 Desember 2007.

[4] Wikipedia.org. *http://en.wikipedia.org*. Tanggal akses 13 Januari 2007.

[5] Symantec Enterprise Security. *Securing Instant messaging White Paper*, 2007.

[6] Yahoo! Inc. *Yahoo! Messenger Plug-in SDK User Guide and Reference*, 2006.