Analisis Keamanan pada GNU Privacy Guard

Anggriawan Sugianto / 13504018
Teknik Informatika - STEI - ITB, Bandung 40132, email: if14018@students.if.itb.ac.id

Abstrak *- GNU Privacy Guard (GnuPG atau GPG) merupakan suatu perangkat lunak open source yang mengimplementasikan standar OpenPGP secara lengkap sebagaimana yang didefinisikan pada RFC4880. Dengan menggunakan GnuPG, pengguna bisa mengenkripsi dan menandatangani data atau pesan dalam komunikasinya. GnuPG menggunakan kombinasi kriptografi kunci-simetrik konvensional dan kriptografi kunci-publik. Layanan keamanan GnuPG meliputi kerahasiaan, manajemen kunci, otentifikasi, dan tanda tangan digital.*

Meskipun GnuPG ini sudah banyak digunakan di berbagai sistem operasi, terutama yang open source seperti Linux, FreeBSD, dan segala varian keduanya, ada beberapa aspek keamanan GnuPG yang patut dipertimbangkan. Salah satu yang perlu diperhatikan adalah keamanan algoritma kriptografi yang digunakan di dalamnya. Pada makalah ini, penulis akan membahas seluk-beluk GnuPG beserta aspek keamanan penggunaannya.

Kata Kunci*: GNU Privacy Guard, enkripsi, tanda tangan digital*

1. PENDAHULUAN

Dewasa ini perkembangan teknologi informasi dan komunikasi sudah berkembang sangat pesat. Hampir di setiap bidang kehidupan telah menggunakan teknologi ini sebagai saran pendukung maupun sarana utama. Sehubungan dengan hal ini, aspek keamanan dalam teknologi informasi dan komunikasi tentunya tidak bisa diabaikan. Dalam kegiatan kirim-terima pesan, aspek keamanan yang perlu diperhatikan antara lain kerahasiaan, integritas data, otentikasi, dan nirpenyangkalan. Aspek keamanan tersebut bisa dijaga dengan memanfaatkan kriptografi.

Berdasarkan penggunaan kuncinya, algoritma kriptografi bisa dibagi menjadi dua, yakni kriptografi kunci-simetri dan kriptografi kunci-asimetri. Kriptografi kunci-simetri menggunakan kunci yang sama (*secret key*) untuk melakukan enkripsi dan dekripsi. Sedangkan kriptografi kunci-asimetri menggunakan kunci yang berbeda untuk melakukan enkripsi dan dekripsi. Pada kriptografi kunci-asimetri, enkripsi dilakukan dengan kunci publik, sedangkan dekripsi dilakukan dengan kunci privat. Karena itu, kriptografi kunci-asimetrik disebut juga kriptografi kunci-publik.

Penggunaan perangkat lunak yang memanfaatkan kriptografi, baik kunci-simetri maupun kunci-asimetri, sangatlah memudahkan pengguna untuk menjaga keamanan komunikasi ataupun data. Dengan memanfaatkan perangkat lunak tersebut, pengguna bisa melakukan enkripsi, tanda tangan digital, ataupun otentifikasi data. Salah satu perangkat lunak yang memanfaatkan kriptografi adalah GNU Privacy Guard (biasa disingkat menjadi GnuPG atau GPG).

2. SEKILAS TENTANG GNU PRIVACY GUARD

GnuPG merupakan perangkat lunak *open source* yang mengimplementasikan standar OpenPGP secara lengkap sebagaimana yang didefinisikan pada RFC4880. Sebagai perangkat lunak *open source*, GnuPG merupakan alternatif dari perangkat lunak PGP (Pretty Good Privacy). GnuPG biasanya sudah tercakup di dalam kebanyakan distro Linux, seperti Debian, MandrakeSoft, RedHat, dan SuSE. Dengan GnuPG, pengguna bisa mengenkripsi dan menandatangani data dan komunikasi. GnuPG merupakan perangkat lunak *command line* yang mudah diintegrasikan dengan aplikasi lainnya. Meskipun demikian, tersedia juga aplikasi tampilan (*frontend*) dan pustaka yang mendukungnya.

Meskipun GnuPG kebanyakan digunakan di lingkungan sistem operasi yang *open souce* seperti Linux dan FreeBSD, kode program GnuPG juga bisa digunakan di lingkungan Windows. Adapun GnuPG versi Windows, dinamakan Gpg4win. Gpg4win ini merupakan bundel program yang dikemas untuk Windows yang mencakup GnuPG, pengelola kunci (WinPT dan GPA), *plugin* untuk enkripsi email (GPGol), *plugin* untuk enkripsi file (GPGee), dan program *email client* (Claws Mail). Pada makalah ini, penulis melakukan eksperimen dengan menggunakan GnuPG versi Windows, atau Gpg4win.

3. CARA KERJA GNU PRIVACY GUARD

Sebagaimana yang telah ditetapkan dalam standar OpenPGP, GnuPG menyediakan layanan integritas pesan dan file data dengan teknologi tanda tangan digital, enkripsi, kompresi, dan konversi Radix-64. GnuPG juga menyediakan layanan manajemen dan sertifikat kunci.

Untuk menjamin kerahasiaan pesan atau file data, GnuPG menggunakan kombinasi kriptografi kunci-

simetrik dan kriptografi kunci-publik. Adapun langkah-langkah menjaga kerahasian data pada pengiriman suatu pesan dengan melakukan enkripsi pada GnuPG adalah sebagai berikut:

1. Pengirim membuat pesan.
2. Pengirim membangkitkan sebuah bilangan acak atau memberikan sandi lewat (*passphrase*) sebagai *session key* untuk pesan saat ini.
3. Pengirim mengenkripsi *session key* tersebut dengan kunci publik masing-masing penerima. Hasil enkripsi *session key* ini menjadi awal dari pesan yang dikirim.
4. Pengirim mengenkripsi pesan (yang biasanya sudah dikompresi) yang akan dikirim dengan menggunakan *session key*.
5. Penerima mendekripsi pesan dengan kunci privatnya.
6. Penerima mendekripsi pesan dengan *session key*. Jika pesan yang diterima merupakan hasil kompresan, maka pesan harus didekompresi.

Baik layanan tanda tangan digital maupun kerahasiaan, bisa diaplikasikan pada pesan yang sama. Caranya, tanda tangan digital dibangkitkan dan dibubuhkan pada pesan. Lalu, pesan dan tanda tangan tersebut dienkripsi dengan menggunakan *session key* yang simetrik. Lalu, *session key* dienkripsi menggunakan enkripsi kunci-publik dan ditaruh di awal blok yang terenkripsi.

Sedangkan langkah-langkah otentifikasi yang dilakukan GnuPG melalui tanda tangan digital adalah sebagai berikut:

1. Pengirim membuat pesan.
2. Pengirim membangkitkan kode *hash* dari pesan.
3. Pengirim membangkitkan tanda tangan digital dari kode *hash* pesan dengan menggunakan kunci privat pengirim.
4. Tanda tangan digital tersebut dilekatkan pada pesan.
5. Penerima menerima pesan yang bertanda tangan.
6. Penerima membangkitkan kode *hash* dari pesan yang diterima dan memverifikasinya dengan menggunakan tanda tangan digital pada pesan. Jika verifikasi berhasil, berarti pesan yang diterima tersebut otentik.

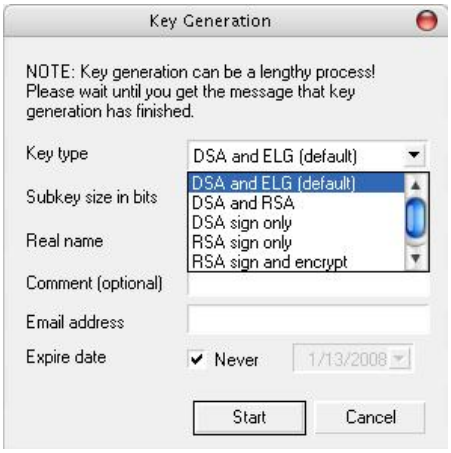
4. ALGORITMA PADA GNU PRIVACY GUARD

Gpg4win versi 1.1.3 yang digunakan untuk eksperimen oleh penulis di dalam makalah ini mengimplementasikan GnuPG versi 1.4.7. GnuPG versi ini mendukung beberapa macam algoritma.

- Algoritma kunci-publik: RSA, RSA-E, RSA-S, ELG-E, DSA
- Algoritma simetrik: 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH
- Algoritma *hash*: MD5, SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224

Dengan bermacam algoritma di atas, pengguna bisa melakukan enkripsi atau memberikan tanda tangan digital dengan algoritma yang berbeda-beda. Meskipun demikian, GnuPG tetap memberikan pilihan *default* bagi pengguna.

Pada Gpg4win, pengguna hanya diberikan sedikit pilihan algoritma. Untuk pembangkitan kunci, diberikan pilihan algoritma: DSA, El Gamal, dan RSA.



Meskipun demikian, pengguna tetap bisa melakukan kustomisasi algoritma melalui perintah baris (*command line*). Misalnya, secara default GnuPG mengeset algoritma CAST5 untuk enkripsi simetrik, jika pengguna mau mengenkripsi suatu file dengan algoritma simetrik Triple DES, maka perintah yang diberikan:

```
gpg --cipher-algo 3DES --symmetric file.ext
```

Pemilihan algoritma yang bagus tentunya sangat mempengaruhi keamanan data.

Selain algoritma kriptografi di atas, GnuPG versi 1.4.7 ini juga mendukung beberapa algoritma kompresi, antara lain: ZIP, ZLIB, BZIP2.

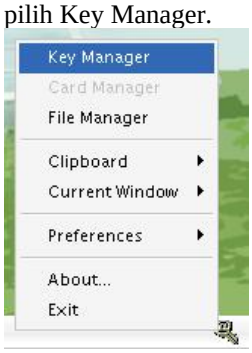
5. EKSPERIMEN TERHADAP GNU PRIVACY GUARD

Pada bagian ini akan dijelaskan mengenai cara penggunaan dan hasil eksperimen menggunakan GnuPG. Beberapa hal yang akan dibahas antara lain cara membuat pasangan kunci, mengirim kunci publik ke server, mendapatkan kunci publik dari server, tanda tangan digital, serta enkripsi dan dekripsi pesan. Semua eksperimen di bawah ini menggunakan GnuPG versi Windows, yaitu: Gpg4win, dengan bantuan WinPT (Windows Privacy Tray) sebagai aplikasi *frontend*-nya. Sebagai alternatif, pengguna bisa menggunakan GPA (GNU Privacy Assistant), yang juga sudah terkandung di dalam bundel Gpg4win.

5.1. Membuat Pasangan Kunci

Berikut ini adalah langkah-langkah untuk membuat pasangan kunci pada GnuPG:

1. Klik Start Menu → All Programs → GnuPG for Windows → WinPT. Setelah itu, akan muncul ikon WinPT pada *System tray*.
2. Klik kanan ikon WinPT pada *system tray*, lalu pilih Key Manager.



Maka muncullah sebuah jendela Key Manager.

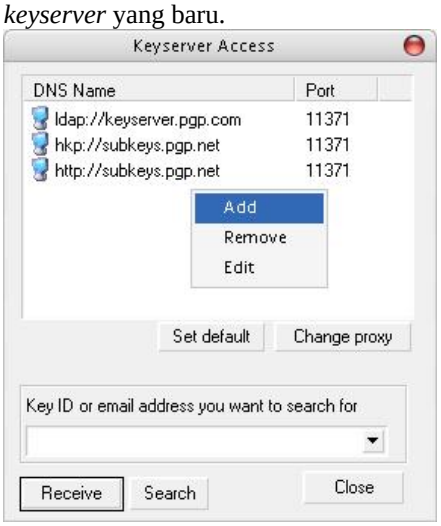
3. Klik menu Key → New → Expert. Maka muncullah form Key Generation.
4. Pilih jenis algoritma yang akan dipakai (misalnya, DSA dan El Gamal sebagaimana pilihan *default*), isikan ukuran kunci, nama, alamat email, dan tanggal habis masa berlaku. Klik Start.

5. Masukkan *passphrase*. Tekan OK. Masukkan *passphrase* yang sama sekali lagi. Tekan OK.
6. Tunggu sampai pasangan kunci terbentuk. Proses pembangkitan pasangan kunci ini cukup lama karena melibatkan perhitungan matematis yang rumit (eksponen modulo).
7. Setelah pasangan kunci dibangkitkan, akan muncul konfirmasi untuk menyimpan *public keyring* dan *secret keyring*. Simpanlah keduanya di tempat yang aman.

5.2. Mengirim Kunci Publik ke Server

Kunci publik perlu dikirim ke server supaya orang lain bisa menggunakannya untuk verifikasi tanda tangan digital, ataupun mengenkripsi pesan yang akan dikirim ke pengguna (yang membangkitkan kunci publik tersebut). Berikut ini adalah langkah-langkah untuk mengirimkan kunci publik ke server:

1. Pastikan *keyserver* sudah terdaftar pada aplikasi. Klik menu Keyserver pada jendela Key Manager. Jika belum ada, klik kanan pada daftar *keyserver* lalu klik Add. Isikan tipe, *hostname*, dan *port* dari *keyserver* yang baru.



2. Kembali ke jendela Key Manager, klik kanan salah satu *key* yang akan dikirim. Pilih Send to Keyserver, lalu pilih *keyserver* tujuan.

5.3 Mendapatkan Kunci Publik dari Server

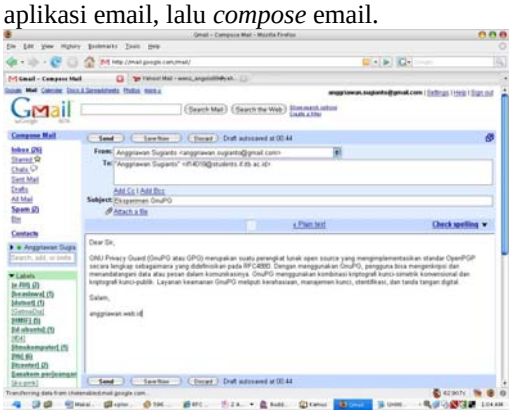
Berikut ini adalah langkah-langkah untuk mendapatkan kunci publik dari server:

1. Klik menu Keyserver pada jendela Key Manager.
2. Pilih dari server mana kunci publik akan didapatkan. Isikan *key ID* atau alamat email, lalu tekan Receive.

5.4. Enkripsi dan Menandatangani Ppesan

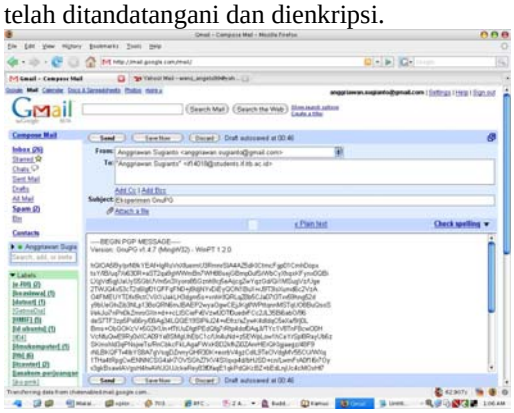
Berikut ini adalah langkah-langkah untuk melakukan enkripsi dan menandatangani pesan:

1. Pesan bisa berupa teks pada aplikasi apapun, baik aplikasi email, aplikasi pengolah kata, maupun teks editor biasa seperti Notepad. Pada eksperimen ini, pengguna ingin mengenkripsi dan menandatangani email. Untuk itu, bukalah aplikasi email, lalu *compose* email.



2. Klik kanan ikon WinPT pada *system tray*. Klik Current Window → Sign & Encrypt

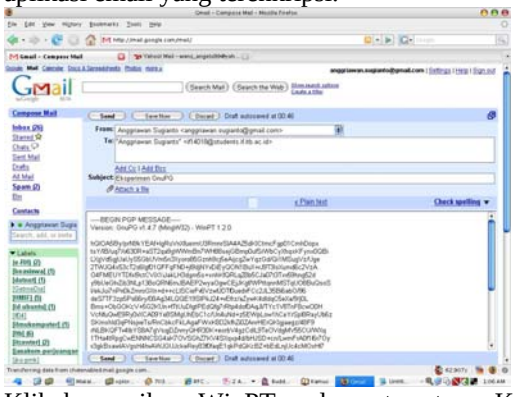
3. Pilih key yang mana untuk enkripsi dan key yang mana untuk tanda tangan digital. Klik OK.
4. Masukkan *passphrase* untuk membuka kunci privat dari key yang dipilih di atas. Klik OK.
5. Lihat aplikasi email, dan perhatikan bahwa pesan telah ditandatangani dan dienkripsi.



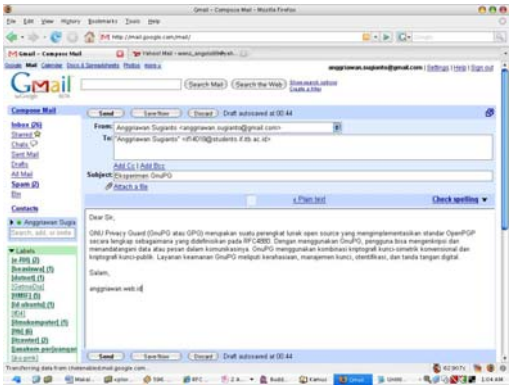
5.5. Dekripsi dan Verifikasi Pesan

Berikut ini adalah langkah-langkah untuk melakukan dekripsi dan memverifikasi pesan:

1. Buka aplikasi yang telah dienkripsi, dalam hal ini aplikasi email yang terenkripsi.



2. Klik kanan ikon WinPT pada *system tray*. Klik Current Window → Decrypt/Verify
3. Masukkan *passphrase* untuk membuka kunci privat dari key yang digunakan. Klik OK. Jika *passphrase* benar, maka akan ada notifikasi bahwa pesan telah diverifikasi.
4. Lihat aplikasi email, dan perhatikan bahwa pesan telah didekripsi menjadi seperti semula.



6. ANALISIS KEAMANAN PADA GNU PRIVACY GUARD

Faktor keamanan sangatlah penting untuk diperhatikan dalam kriptografi. Sekalipun telah menggunakan perangkat lunak yang populer seperti GnuPG, pengguna tetap harus waspada terhadap aspek keamanannya. Perangkat lunak biasanya memiliki *bug* atau kesalah kode program yang terkadang bisa memberikan celah keamanan. Selain itu, algoritma kriptografi yang digunakan oleh suatu perangkat lunak juga bisa menjadi celah keamanan jika ternyata kelemahan algoritma tersebut telah ditemukan.

Pada bagian ini akan dibahas beberapa kelemahan yang pernah ditemukan pada GnuPG.

6.1. Masalah Pesan Ganda

Pada awal tahun 2007, Gerardo Richarte, dari Core Security Technologies, menemukan masalah ketika menggunakan GnuPG pada mode *streaming*. Karena adanya *bug* tersebut, maka penyerang bisa menyisipkan teks tambahan sebelum atau sesudah tanda tangan digital. Dengan demikian, pengguna akan mengira bahwa teks tambahan tersebut termasuk tanda tangan digital. *Bug* ini telah diperbaiki dengan mengubah GnuPG sehingga pesan OpenPGP palsu dapat terdeteksi dan verifikasi akan gagal. Perbaikan ini diikutsertakan pada GnuPG 1.4.7.

6.2. Pointer Fungsi yang Bisa Diatur

Pada tahun 2006, Tavis Ormandy, dari tim keamanan Gentoo, menemukan *bug* yang bisa dieksploitasi pada pemrosesan paket yang terenkripsi pada GnuPG. Dengan memanfaatkan paket OpenPGP yang cacat, penyerang bisa mengubah referensi dari pointer fungsi pada GnuPG. Hal ini merupakan *bug* yang sangat kecil untuk dieksploitasi. *Bug* ini mempengaruhi penggunaan GnuPG di mana penyerang bisa mengatur data yang diproses GnuPG. Hal ini tidak terbatas pada data yang terenkripsi, tetapi data yang ditandatangani pun bisa dipengaruhi. *Bug* ini telah diperbaiki dengan dirilisnya GnuPG 1.4.6 dan *patch* untuk GnuPG 2.0.1.

6.3. *Buffer Overflow*

Bug ini ditemukan pada GnuPG 1.4 dan 2.0. Saat menjalankan GnuPG secara interaktif, pesan-pesan tertentu bisa digunakan untuk membuat GPG mengalami *crash*. Sejak versi 1.4.7 dan 2.0.3, bug ini telah diperbaiki.

6.4. Verifikasi Tanda Tangan *False Positive*

Pada awal tahun 2006, Gentoo project menemukan bug pada GnuPG, yaitu mungkin terjadinya verifikasi tanda tangan digital yang *false positive*. Tanda tangan digital yang seharusnya benar, ternyata gagal diverifikasi. Verifikasi yang *false positive* ini hanya terjadi pada tanda tangan digital yang tidak dilekatkan pada pesan, yang dilakukan oleh *script* atau program email. Penggunaan GnuPG dengan cara interaktif tidak terpengaruh oleh *bug* ini. *Bug* ini telah diperbaiki pada GnuPG 1.4.2.1

6.5. GnuPG Tidak Mendeteksi Injeksi Data Tidak Bertandatangan

Bug ini muncul akibat kesalah saat memperbaiki *bug* verifikasi *false positive* sebelumnya. Karena *bug* ini, verifikasi tanda tangan yang dilekatkan pada pesan bisa memberikan hasil yang positif, tapi saat saat mengekstrak datayang tidak bertandatangan, data tersebut mungkin ditambahi data tambahan di luar tanda tangan digital. Karena itu, hal ini memungkinkan penyerang untuk mengambil pesan yang bertandatangan, lalu memasukkan data tambahan yang sewenang-wenang. *Bug* ini telah diperbaiki paa GnuPG 1.4.2.2

6.6. Kelemahan Kunci ElGamal Ditemukan

Pada tahun 2003, Phong Nguyen menemukan *bug* pada cara GnuPG membuat dan menggunakan kunci ElGamal untuk tanda tangan digital. Hal ini merupakan kegagalan keamanan yang signifikan yang bisa menyebabkan hampir semua kunci ElGamal yang digunakan untuk menandatangani bisa dipecahkan dalam hitungan detik. Semua kunci ElGamal (tipe 20) untuk tanda tangan dan enkripsi mungkin untuk dipecahkan. Solusi yang diberikan GnuPG adalah untuk tidak menggunakan kunci ElGamal (tipe 20) untuk tanda tangan + enkripsi. Pada GnuPG versi selanjutnya (setelah GnuPG 1.0.2), kemampuan GnuPG untuk membuat kunci tanda tangan digital + enkripsi ElGamal dihilangkan.

7. KESIMPULAN DAN SARAN

7.1. Kesimpulan

Berdasarkan uraian yang telah penulis sampaikan di atas, bisa ditarik beberapa kesimpulan bahwa GNU Privacy Guard merupakan perangkat lunak kriptografi yang cukup baik. Fitur-fitur yang ada memudahkan pengguna untuk bisa menjaga kerahasiaan dan otentifikasi pesan. Ketersediaan beberapa algoritma di dalam GnuPG juga membuat pengguna bisa lebih leluasa untuk mengatur kerahasiaan pesannya.

Ditinjau dari aspek keamanan, GnuPG cukup kuat, dibuktikan dengan belum adanya metode kriptanalisis yang ditemukan untuk memecahkan enkripsi GnuPG sampai saat ini. Hal ini didukung dengan penggunaan algoritma kriptografi yang kuat, seperti RSA, DSA, dan algoritma lainnya yang memerlukan perhitungan yang cukup rumit.

Mengenai masalah keamanan program, untuk menemukan kelemahan perangkat lunak (berupa *bug*) tidaklah mudah karena harus mendalami setiap potongan kode yang menyusun program tersebut. Untuk setiap *bug* GnuPG yang ditemukan, pihak pengembang biasanya selalu segera memberikan solusinya berupa merilis *patch* untuk versi lama ataupun merilis program versi baru yang bebas dari *bug* tersebut. Hal ini tentunya memudahkan pengguna untuk terus memantau keamanan perangkat lunak kriptografi yang digunakannya.

7.2. Saran

Saran yang penulis sampaikan terkait keamanan pada GNU Privacy Guard, yakni pengguna sebaiknya selalu waspada terhadap kemungkinan adanya *bug* pada setiap perangkat lunak, termasuk GnuPG. Dan jika menemukan *bug* tersebut, sebaiknya segera dipublikasikan, terutama ke pihak pengembang, supaya *bug* tersebut segera diatasi sehingga keamanan pesan tersebut bisa terus terjaga.

DAFTAR REFERENSI

[1] http://www.gnupg.org/
[2] http://tools.ietf.org/rfc/rfc4880.txt
[3] Ashley, Mike, *The GNU Privacy Handbook*, The Free Software Foundation 1999
[4] Heinze, Manfred J., *Gpg4win for Novices*, The Free Software Foundation 2006
[5] Koch, Werner, *Using The GNU Privacy Guard*, The Free Software Foundation 2007
[6] Menezes,Alfred, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press 1996
[7] Munir, Rinaldi, *Diktat Kuliah IF5054 Kriptografi*, STEI ITB 2007