Penerapan Algoritma Kriptografi Kunci Publik untuk *Repository* Organisasi

Emil Fahmi Yakhya - 13509069[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]emilfahmi@students.itb.ac.id

Abstrak – Algoritma kunci publik merupakan teknik kriptografi asimetris dimana kunci yang digunakan untuk dekripsi dan enkripsi berbeda. Pada makalah ini, dibahas konsep algoritma kunci publik, contoh penerapan dan juga studi kasusnya. Penggunaan fasilitas-fasilitas IT di masa kini semakin rawan untuk disalahgunakan dan perlu ditingkatkan keamanannya dengan cara memanfaatkan algoritma kriptografi kunci publik. Studi kasus yang dijadikan contoh dalam penerapan di makalah ini adalah repository yang dimiliki oleh sebuah organisasi dimana repository tersebut digunakan oleh banyak orang dengan identitas yang berbeda-beda.

Kata kunci – Kunci Publik, Kunci Privat, Repository

I. PENDAHULUAN

Kriptografi adalah kata serapan dari bahasa asing, dalam hal ini bahasa inggris, yaitu cryptography. Cryptography atau cryptology berasal dari bahasa Yunani, yaitu κρσπτός, kryptos, "hidden, secret" atau “tersembunyi, rahasia”, dan γράφω, gráphō, "I write" atau “aku menulis”, dan -λογία, -logia atau “ilmu”. Kriptografi adalah ilmu atau seni untuk menyembunyikan informasi. Proses menyembunyikan informasi ini dilakukan dengan teknik penyandian, atau mengubah pesan atau informasi menjadi sandi-sandi yang tidak dimengerti oleh orang lain, selain pembuat dan penerimanya.

Pada kriptografi dikenal istilah-istilah seperti plain text, cipher text, enkripsi, dan dekripsi. Plain text adalah pesan asli yang ingin dikirimkan. Cipher text adalah pesan yang telah disandikan dengan metode enkripsi tertentu. Enkripsi adalah proses mengubah sebuah plain text menjadi cipher text, dan dekripsi adalah proses mengubah sebuah cipher text menjadi plain text.

Dalam kehidupan sehari-hari, terdapat banyak transaksi data dilakukan. Beberapa informasi yang dikirimkan dalam transaksi tersebut adalah informasi-informasi yang bersifat rahasia dan pribadi. Karena itu data-data yang dikirimkan perlu dirahasiakan sehingga pihak lain yang mencoba mendapatkan informasi tersebut tanpa izin tidak akan dapat mengetahuinya.

Di tahun 1976, sebuah sistem kriptografi yang asymmetric-key dipublikasikan oleh Whitfield Diffie dan Martin Hellman yang dipengaruhi oleh metode distribusi kunci publik (public-key) yang dikembangkan oleh Ralph Merkle. Sistem pertukaran kunci ini dikenal dengan nama Diffie-Hellman key exchage (pada tahun 2002, Hellman menyarankan agar nama ini diubah menjadi Diffie-Hellman-Merkle key exchage atas jasa Merkle dalam penemuan metode public-key). Sistem ini juga disebut public-key cryptography. Dalam public-key cryptography, kedua pihak yang ingin mengirimkan dan menerima pesan akan saling bertukar kunci yang digunakan untuk mengenkripsi data, sedangkan kunci untuk melakukan dekripsi tetap dirahasiakan.

Di masa kini, algoritma kriptografi kunci public telah diterapkan secara luas di kehidupan sehari-hari. Mulai dari penggunaan ATM, media sosial, hingga penggunaan oleh pribadi atau organisasi. Dalam makalah ini, akan dibahas mengenai bagaimana penerapan algoritma kunci publik di organisasi dan mengambil contoh studi kasus *repository* di HMIF ITB.

II. ALGORITMA KUNCI PUBLIK

All printed material, including text, illustrations, and charts, must be kept within a print area of 17 cm wide by 25 cm high. Do not write or print anything outside the print area. All *text* must be in a two-column format. Columns are to be 8.25 cm wide, with a 0.5 cm space between them. Text must be fully justified.

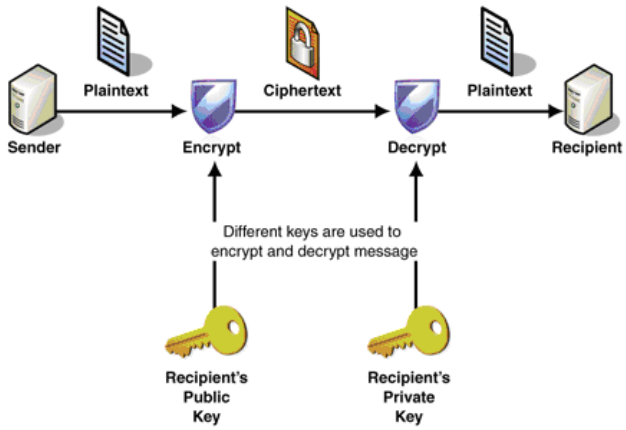
Algirutma kriptografi kunci publik disebut juga sebagai algoritma kriptografi kunci-nirsimitri. Pada kriptografi kunci public, masing-masing pengirim dan penerima mempuanyai sepasang kunci:

1. Kunci public: untuk mengenkripsi pesan
2. Kunci privat: untuk mendekripsi pesan.

$$E_e(m) = c \text{ dan } D_d(c) = m$$

Berikut adalah skema algoritma kriptografi kunci

publik:



Sumber: http://clean-clouds.com/2011/03/08/encryptionhybrid-approach/

Gambar 1 Skema Kunci Publik

Misalkan: Alice sebagai pengirim pesan dan Bob sebagai penerima pesan. Alice mengenkripsi pesan dengan kunci publik Bob, kemudian Bob akan mendekripsi pesan dengan kunci privatnya (kunci privat Bob). Sebaliknya, Bob mengekripsi pesan dengan kunci publik Alice dan Alice akan mendekripsi pesan dengan kunci privatnya (kunci private Alice). Dengan mekanisme tersebut, tidak ada kebutuhan untuk mengirimkan kunci rahasia (seperti halnya pada sistem kriptografi simetri).

Berikut adalah dua algoritma kriptografi kunci publik yang digunakan secara luas:

A. Algoritma RSA

RSA adalah salah satu contoh kriptografi yang menerapkan konsep public key. Algoritma ini pertama kali dipublikasikan di tahun 1977 oleh Ron Rivest, Adi Shamir, dan Leonard Adleman dari Massachusetts Institute of Technology (MIT). Nama RSA sendiri adalah singkatan dari nama belakang mereka bertiga.

Clifford Cocks, seorang matematikawan Inggris sebenarnya juga telah mengembangkan algoritma yang hampir sama dengan RSA ini pada tahun 1973. Namun algoritma buatannya tidak begitu dikenal oleh publik, dan baru dipublikasi pada tahun 1997 karena merupakan proyek rahasia. Walau begitu algoritma yang dikembangkan Rivest, Shamir, dan Adleman tidak berhubungan dengan pekerjaan Cocks.

Pada algoritma RSA terdapat 3 langkah utama yaitu key generation (pembangkitan kunci), enkripsi, dan dekripsi.

Kunci pada RSA mencakup dua buah kunci, yaitu public key dan private key. Public key digunakan untuk melakukan enkripsi, dan dapat diketahui oleh orang lain. Sedangkan private key tetap dirahasiakan dan digunakan untuk melakukan dekripsi.

Pembangkitan kunci atau key generation dari RSA adalah sebagai berikut :

1. Pilih dua buah bilangan prima sembarang a dan b. Kedua bilangan ini tidak boleh diketahui oleh orang lain
2. Hitung n = a x b. Hasil dari perkalian ini tidak perlu dirahasiakan dari pihak luar
3. Hitung m = a – 1 x (b – 1). Setelah *m* dihitung maka kedua bilangan, a dan b, sudah dapat dibuang (dihapus) untuk menjaga kerahasian dari bilangan tersebut
4. Pilih sebuah bilangan bulat untuk kunci publik, sebut namanya *e*, yang relatif prima terhadap *m* (relatif prima).
5. Hitung kunci dekripsi, d, dengan kekongruenan

$$ed \equiv 1 \pmod{m}$$

Dari hasil pembangkitan kunci didapat kunci publik, *e*, dan kunci privat, *d*. Seperti yang sudah dijelaskan sebelumnya kunci publik digunakan untuk mengenkripsi suatu pesan dengan menggunakan rumus

$$C_i = P_i^e \bmod n$$

Sedangkan untuk proses dekripsi menggunakan rumus

$$P_i = C_i^d \bmod n$$

Blok-blok plainteks dinyatakan dengan p_1, p_2, p_3, …, *pn* (harus dipenuhi persyaratan bahwa nilai *pn* harus terletak dalam himpunan nilai 0, 1, 2, ..., n-1 untuk menjamin hasil perhitungan tidak berada di luar himpunan). Pada langkah kelima pembangkitan kunci atau *key generation*, kekongruenan $ed \equiv 1 \bmod m$sama dengan $ed \bmod m \equiv 1$. Sehingga dapat pula dikatakan bahwa $ed \equiv 1 \pmod{m}$ ekivalen dengan $ed = 1 + km$. Maka *d* dapat dihitung dengan cara yang sederhana dengan persamaan

$$d = \frac{1 + km}{e}$$

Dengan mencoba nilai k = 1, 2, 3, …, diperoleh nilai d yang bulat. Nilai *d* inilah yang dirahasiakan sebagai kunci privat, yang digunakan untuk mendekripsi suatu pesan yang sudah dienkripsi menggunakan kunci publik *e*.

Dalam implementasi bear dari nilai a dan b adalah suatu bilangan prima yang lebih besar dari 100 digit. Hal ini diperlukan agar untuk memfaktorkan hasil perkalian dibutuhkan waktu yang sangat lama untuk menemukan kedua bilangan tersebut.

Kekuatan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi faktor primanya, dalam hal ini memfaktorkan n menjadi *a* dan *b*. Karena ketika *n* berhasil difaktorkan, maka dapat diketahui nilai

m. Meskipun nilai *e* diumumkan, perhitungan kunci *d* tidaklah mudah karena nilai *m* yang tidak diketahui.

B. Algoritma ElGamal

Algoritma Elgamal merupakan salah satu algoritma kriptografi kunci-publik yang dibuat oleh Taher ElGamal pada tahun 1984. Algoritma in pada umumnya digunakan untuk digital signature, namun kemudian dimodifikasi sehingga juga bisa digunakan untuk enkripsi dan deskripsi. ElGamal digunakan dalam perangkat lunak sekuriti yang dikembangkan oleh GNU, program PGP, dan pada sistem sekuriti lainnya. Kekuatan algoritma ini terletak pada sulitnya menghitung logaritma diskrit.

Algoritma Elgamal tidak dipatenkan. Tetapi, algoritma ini didasarkan pada algoritma Diffie – Hellman, sehingga hak paten algoritma Diffie – Hellman juga mencakup algoritma ElGamal. Karena hak paten algoritma Diffie – Hellman berakhir pada bulan April 1997, maka algoritma ElGamal dapat diimplementasikan untuk aplikasi komersil.

Besaran-besaran yang digunakan dalam pembangkitan kunci publik algoritma ElGamal:

1. Bilangan prima, *p* (tidak rahasia)
2. Bilangan acak, *g* ($g < p$) (tidak rahasia)
3. Bilangan acak, *x* ($x < p$) (rahasia)
4. Blok plainteks M (plainteks) (rahasia)
5. *a* dan *b* (cipherteks) (tidak rahasia)

Setelah bilangan *p*, *g*, dan *x* diketahui pada untuk membangkitkan kunci publik dapat dilakukan dengan rumus

$$y = g^x mod\ p$$

Maka didapat kunci publik *y* dan kunci privat *x*, bilangan *p* dan *g* harus diberikan kepada pihak luar agar dapat mengenkripsi pesan yang akan dikirim.

Sebelum melakukan enkripsi pesan dibagi menjadi m_1, m_2, m_3, …, m_n dengan syarat besar dari m_n harus lebih kecil dari *p* dan tidak boleh negatif. Untuk melakukan enkripsi, dilakukan tahap-tahap sebgai berikut:

1. Pilih bilangan acak *k*, yang lebih kecil dari $p - 1$, dan relatif prima dengan $p - 1$.
2. Hitung bilangan *a* dan *b* dengan menggunakan rumus

$$a = g^k\ mod\ p$$

$$b = y^k m\ mod\ p$$

Untuk melakukan proses dekripsi maka dilakukan tahap-tahap sebagai berikut:

1. Hitung nilai *s*, dengan rumus $s = a^x\ mod\ p$
2. Melakukan dekripsi pada pesan dengan menggunakan rumus

$$m = bs^{-1}\ mod\ p$$

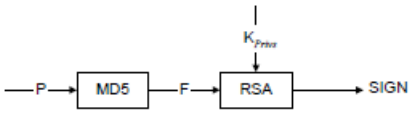
III. Penerapan Algoritma Kriptografi Kunci Publik

Contoh penerapan dari algoritma Kriptografi Kunci Publik adalah PGP (*Pretty Good Privacy).* PGP dahulunya digunakan untuk mengamankan pengiriman pesan/email, namun kini PGP juga dapat digunakan untuk mengamankan berbagai file dan program pada komputer personal.

Cara kerja PGP terbagi menjadi lima operasi utama yaitu:

a. Otentikasi

Selain mengenkripsi pesan / email dan file, PGP juga dapat memberi tanda tangan digital. Adapun tujuan dari proses tersebut adalah sebagai otentikasi dari pesan / file yang bertujuan untuk memastikan / mengecek apakah file tersebut masih asli atau sudah diubah oleh orang lain atau bisa juga sudah terserang oleh virus ataupun trojan.

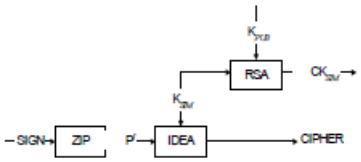


Gambar 1 Tanda tangan digital

Dari gambar diatas terlihat pesan (P) dimasukkan kedalam fungsi MD5 yang menghasilkan sidik jari (F). Sidik jari tersebut adalah identitas pesan. Sidik jari (F) ditanda tangani oleh pengirim (sumber S) menggunakan kunci privat K_{priv} S . Hasil tandatangan SIGN akan digunakan pada bagian enkripsi.

b. Kerahasiaan

PGP menggunakan algoritma IDEA (*International Data Encryption Algorithm*) dengan kunci 128 bit untuk menyandikan data (pesan / file) dan menggunakan mode *Chiper Fedd Back* (CFB) dengan menggunakan vector awal (*Initialization Vector IV*) nol. Hal yang perlu diperhatikan adalah distribusi kunci dalam PGP, setiap kunci konvensional hanya digunakan sekali. Artinya setiap kali ada pesan yang akan dienkrip dibangkitkan kunci baru 128 bit secara acak. Meskipun dalam dokumentasi kunci ini disebut sebagai kunci rahasia / *simetri*. Karena digunakan Cuma sekali maka kunci *simetri* ini digabungkan dengan pesan yang sudah dienkrip dengan kunci tersebut kemudian dikirim bersama-sama. Dan untuk melindunginya, kunci *simetri* tersebut dienkripsi dengan kunci publik penerima.



Gambar 2 Enkripsi pada PGP

Pesan P digabungkan dengan tanda tangan pesan SIGN, dikompres untuk mengurangi karakter berulang sehingga lebih mempersulit *cryptanalyst* untuk membongkar *chiper*nya. Kompresi juga dimaksudkan untuk mengurangi ukuran file sebagai akibat membesarnya file akibat operasi BASE-64.

Hasil kompresi P^I dienkrip oleh fungi IDEA dengan kunci simetri K_{SIM} sehingga menghasilkan cipher.

Kemudian K_{SIM} ini dienkrip oleh RSA (nama penemu algoritma tersebut ***R**on Rivest, Adi **S**hamir and Leonard **A**dleman*) supaya dapat dikirimkan ke tujuan. Kunci yang digunakan untuk mengenkrip K_{SIM} adalah K_{PUB} hasil keluarannya adalah CK_{SIM}

c. Kompresi

PGP akan mengompres pesan setelah ditandatangan dan sebelum enkripsi. Alasan kompresi ini ditujukan untuk memperkuat keamanan kriptografi karena pesan yang dikompres memiliki redundansi yang lebih sedikit dibanding plainteks aslinya sehingga analisis terhadap cipher akan lebih sulit dilakukan.

d. Kompabilitas E-mail

Ketika PGP digunakan, paling sedikit terdapat satu blok terkirim yang dienkrip. Jika layanan yang digunakan hanya tandatangan digital, maka *message digest* dienkripsi (menggunakan kunci privat RSA pengirim). Bila layanan yang digunakan mengenkripsi pesan dan tandatangan (dengan kunci IDEA), sebagian atau seluruh blok yang dihasilkan PGP terdiri dari aliran 8-bit. Aliran bit ini nantinya akan dikonversi kembali menjadi karakter ASCII yang dikenali.

e. Segmentasi

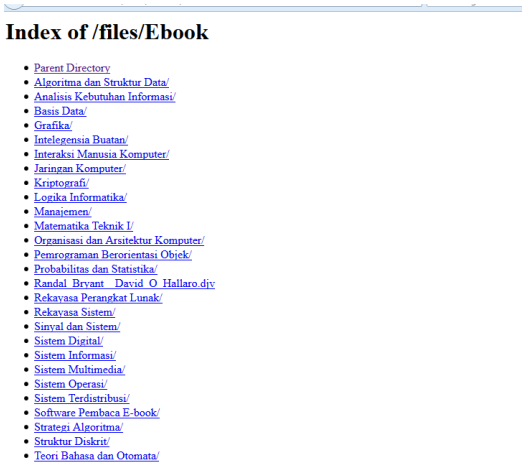
Fasilitas email membatasi panjang maksimum pesan. Banyak fasilitas di internet yang hanya dapet menerima pesan dengan panjang 50.000 octets (kumpulan 8-bit). Untuk mengakomodasi keterbatasan tersebut, PHP secara otomatis melakukan *subdivides* pesan yang besar menjadi beberapa segmen yang besarnya cukup untuk dikirim via email.

Di dalam PGP, ada empat macam kunci yang digunakan antara lain:

1. Kunci konvensional simetri satu waktu (*one time key)*
   Algoritma kriptografi yang digunakan adalah IDEA dengan tujuan mengenkripsi pesan untuk dikirimkaan. Setiap kunci simentri hanya digunakan sekali dan dibangkitkan secara acak.
2. Kunci Publik
   Algoritma kriptografi yang digunakan adalah RSA dan digunakan untuk mengenkripsi kunci simetri untuk dikirimkan bersama pesan. Pengirim dan penerima harus memiliki kunci publik antar sesama pengguna.
3. Kunci Privat
   Algoritma kriptografi yang digunakan adalah RSA dan digunakan untuk mengenkripsi sidik jari pesan untuk membentuk tanda tangan digital. Kunci privat hanya dimiliki oleh pengguna.
4. Kunci Turunan Passphrase
   Algoritma kriptografi yang digunakan adalah IDEA, digunakan untuk mengenkripsi kunci privat

V. Studi Kasus Dan Penerapan Algoritma

Studi kasus pada makalah ini dilakukan dengan cara mengadaptasi konsep PGP pada repository yang dimiliki oleh HMIF ITB (http://hmif.itb.ac.id) dengan cara membuat prototype aplikasi.



Index of /files/Ebook

- Parent Directory
- Algoritma dan Struktur Data/
- Analisis Kebutuhan Informasi/
- Basis Data/
- Grafika/
- Intelegensia Buatan/
- Interaksi Manusia Komputer/
- Jaringan Komputer/
- Kriptografi/
- Logika Informatika/
- Manajemen/
- Matematika Teknik I/
- Organisasi dan Arsitektur Komputer/
- Pemrograman Berorientasi Objek/
- Probabilitas dan Statistika/
- Randal_Bryant__David_O_Hallaro.djv
- Rekayasa Perangkat Lunak/
- Rekayasa Sistem/
- Sinyal dan Sistem/
- Sistem Digital/
- Sistem Informasi/
- Sistem Multimedia/
- Sistem Operasi/
- Sistem Terdistribusi/
- Software Pembaca E-book/
- Strategi Algoritma/
- Struktur Diskrit/
- Teori Bahasa dan Otomata/

Berikut adalah langkah pengaksesan *repository* HMIF ITB:

1. Buka alamat website *repository*.
2. Akses fitur yang diinginkan
3. Keluar dari *repository*.

Dari langkah-langkah di atas, terdapat masalah yakni *repository* dapat diakses oleh seluruh pengguna yang mengetahui alamat *repository*. Dalam kasus ini, bisa saja terdapat penyalahgunaan atau pengerusakan terhadap berkas-berkas yang dimiliki HMIF.

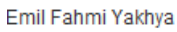
Untuk mengatasi hal tersebut, dapat dilakukan dengan cara membuat mekanisme otentikasi bagi pengguna yang

mengakses *repository*. Mekanisme otentikasi ini dilakukan dengan menggunakan algoritma kriptografi kunci publik ElGamal. Otentikasi akan dilakukan sebelum langkah pengaksesan fitur dilakukan.

Berikut adalah proses autentikasi pada *repository*:

1. Pengguna akan diberikan nilai random P dari sever.
2. Pengguna memasukkan nilai G yang merupakan NIM masing-masign anggota.
3. Nilai X adalah password berupa nilai ASCII yang telah dikonversi yang hanya diketahui oleh anggota HMIF yang menggunakan *repository*.
4. Setiap pengguna yang akan mengupload file harus memasukkan NIM-nya dan kunci publik X (password) kepada Server.
5. Setiap file yang diupload akan dienkripsi dengan kunci publik yang telah diberikan, begitu pula setiap file yang didownload harus didekripsi terlebih dahulu. Enkripsi dan dekripsi ini hanya berhasil apabila pengguna memasukkan NIM yang terdaftar di dalam *database* dan password yang benar.

Gambar 5 – Contoh file terenkripsi



Gambar 6 – Contoh file asli

Dalam *prototype* aplikasi yang dibuat berhasil dilakukan enkripsi terhadap file-file yang akan disimpan dengan cara mengkombinasikan NIM terdaftar dan password yang dikonversi menjadi bentuk ASCII. Mekanisme memanfaatkan algoritma ElGamal untuk mengkombinasikan data pengguna dengan password bersama bisa diterapkan kepada *repository* yang dimiliki suatu organisasi.

VI. KESIMPULAN

Berikut adlaah beberapa kesimpulan yagn dapat diberikan dari penerapan algoritma kriptografi kunci publik apda organisasi:

1. Algoritma kriptogtafi kunci publik memiliki tingkat keamanan yang lebih tinggi dibandingkan algoritma simetri karena proses enkripsi yang lebih kompleks dengan berbagai kombinasi kunci.
2. Penerapan Algoritma kunci publik di masa kini tidak hanya pada email saja, namun sudah bisa diterapkan pada berbagai teknis kehidupan sehari-hari.
3. PGP merupakan salah satu pengembangan algoritma kriptografi kunci publik yang dapat diadaptasi pada berbagai jenis aplikasi.
4. Algoritma Kriptografi kunci publik dapat diterapkan pada *repository* (tempat penyimpanan data) organisasi. Hal ini ditunjukkan dengan berhasilnya mengkombinasikan NIM terdaftar dengan password bersama sebagai kunci publiknya.
5. Penerapan lebih lanjut terhadap algoritma kriptografi kunci publik akan sangat bergantung terhadap *feasibility* dari *platform* yang dimiliki oleh pemangku kepentingan. Misalkan dalam studi kasus bila ingin diterapkan lebih lanjut harus mendapatkan akses khusus terhadap database yang digunakan.

VII. UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Allah yang Maha Kuasa dengan segala anugerah-Nya yang diberikan penulis dapat mengaplikasikan ilmu ang didapatkan di kepas pada dunia nyata. Tak lupa penulis juga mengucapkan terima kasih kepada Bapak Rinaldi Munir sebagai dosen pengajar kuliah IF3058 atas segal ilmu yang telah diberikan hingga makalah ini dapat disempurnakan. Tak lupa pula, terima kasih diucapkan kepada rekan Kerja Praktek Penulis yang telah memberikan referensi terkait PGP (*Pretty Good Privacy*) yang metode otentikasinya dapat diadaptasi dalam berbagai aplikasi.

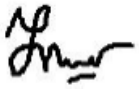
REFERENSI

[1] Slide Kuliah IF3058 – Kriptografi
[2] Munir, Rinaldi. 2006. Kriptografi. Bandung: Informatika.
[3] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. "Chapter 8 Public Key Encryption". Handbook of Applied Cryptography. CRC Press.
[4] http://agcrypt.wordpress.com/2008/02/25/elgamal-algorithm/
[5] http://www.pgp.org
[6] http://www.pgpi.com

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, 20 Mei 2013



Emil Fahmi Yakhya - 1350969