

Kriptografi dan Implementasinya pada Email menggunakan GnuPG

Lukman HDP/s3trum (lukman@ugm.ac.id)

Oktober 2, 2005

Tulisan ini bertujuan memberi pemahaman singkat tentang apa itu kriptografi, mengapa harus menggunakan kriptografi dan bagaimana implementasinya pada sistem komunikasi berbasis email. Akan dicontohkan penggunaan pada MUA Mozilla Thunderbird (menggunakan enigmail) dan pada MS Outlook (menggunakan GPGShell). Tidak ada copyright apapun dalam dokumen ini, anda bebas menyalin, mencetak, maupun memodifikasi (dengan menyertakan nama penulis asli). Kritik, koreksi, saran dan lain-lain silahkan dialamatkan ke email tersebut di atas. Semoga bermanfaat.

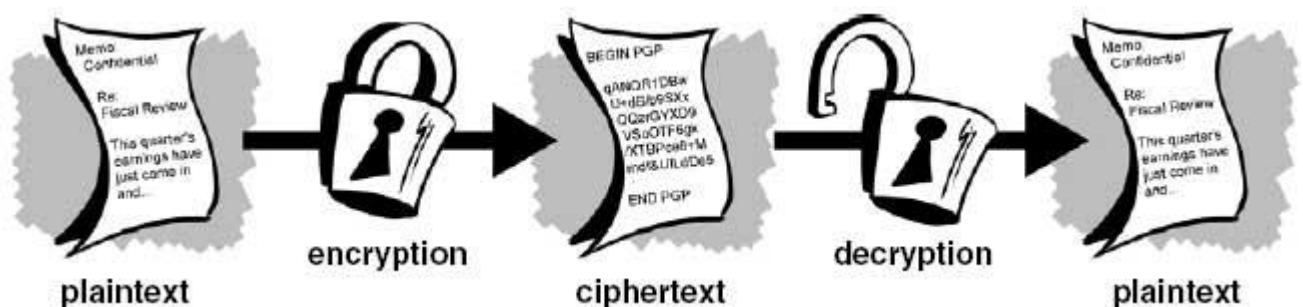
Saat Julius Caesar (tahun berapa ya??) mengirim pesan kepada salah seorang Jendral-nya, dia tidak mempercayai kurirnya, sehingga dia mengganti setiap huruf A dengan D, setiap huruf B dengan E, dan seterusnya dalam keseluruhan alfabet. Hanya orang yang mengetahui rumus "shift by 3" tersebut yang bisa membaca pesannya.

Begitulah enkripsi kita dimulai.

A. Kriptografi

1. Enkripsi dan Dekripsi

Data pada umumnya seperti yang bisa kita baca tanpa ada perlakuan khusus disebut sebagai *cleartext* atau *plaintext*. Cara/metode yang digunakan untuk menyamarkan isi data (seperti yang dilakukan Julius Caesar di atas) disebut sebagai *enkripsi*. Data yang sudah mengalami proses enkripsi sehingga tidak bisa dibaca langsung disebut sebagai *ciphertext*. Sedangkan proses membalik dari ciphertext menjadi plaintext kembali disebut sebagai *dekripsi*.

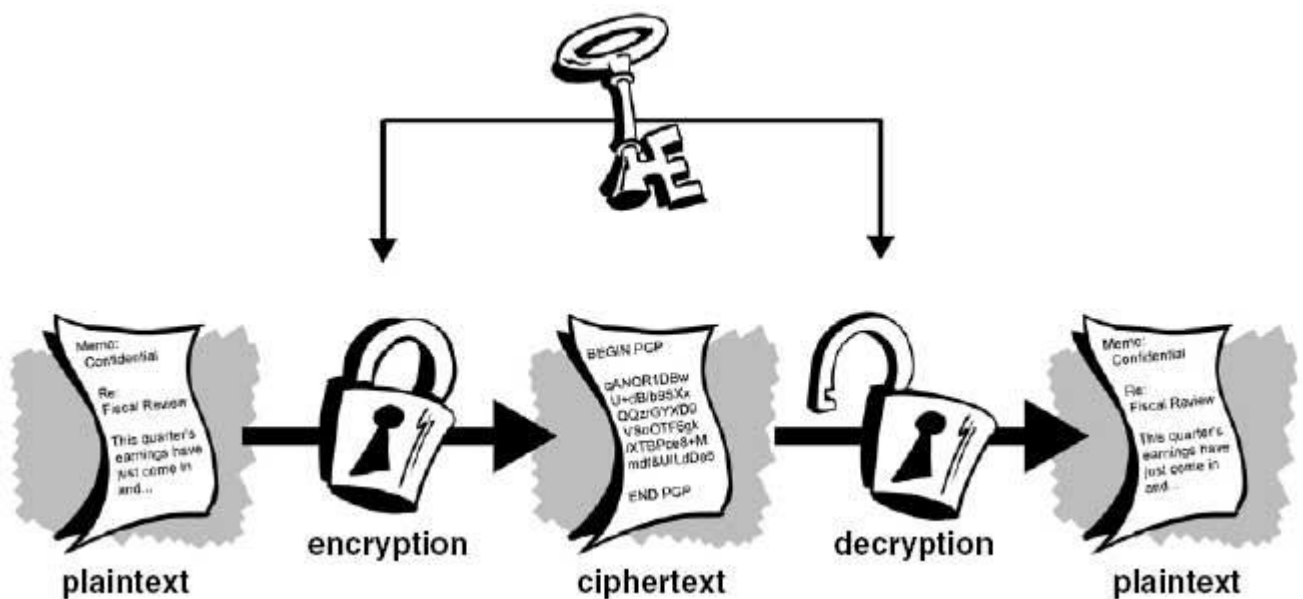


Sedangkan *Kriptografi* adalah ilmu menggunakan persamaan matematika untuk melakukan enkripsi dan dekripsi data. Dengan kriptografi kita bisa mengirimkan data melalui jaringan yang tidak aman (internet) sehingga hanya penerima yang memang dituju yang bisa membaca informasi yang kita kirimkan.

Kriptografi terdiri atas dua jenis yaitu *symmetric cryptography* dan *asymmetric cryptography*.

2. Symmetric cryptography

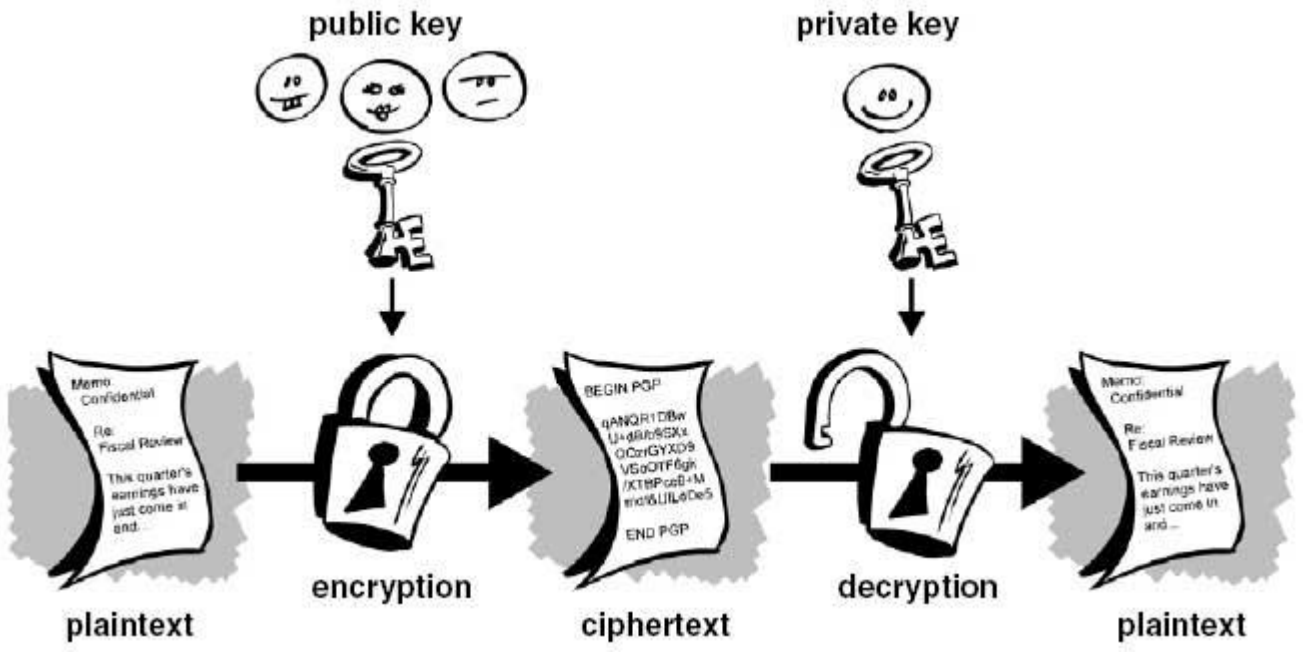
Kriptografi menggunakan kunci simetris disebut juga sebagai kriptografi konvensional. Pada kriptografi jenis ini, satu kunci yang sama digunakan baik untuk melakukan enkripsi maupun dekripsi. Contoh yang paling sederhana adalah metode yang digunakan oleh Julius Caesar seperti yang disebut di atas. Data dienkripsi menggunakan kunci "shift by 3". Dan untuk melakukan dekripsi, digunakan kunci yang sama "shift by 3" juga, hanya bentuknya mungkin berupa fungsi invers-nya (*shift right by 3* dan *shift left by 3*). Contoh lain adalah DES (*Data Encryption Standard*) yang disempurnakan menjadi AES (*Advanced Encryption Standard*).



3. Asymmetric cryptography

Kriptografi jenis ini menggunakan sepasang *key* untuk melakukan proses enkripsi dan dekripsi. Proses Enkripsi dilakukan menggunakan suatu *key* yang disebut sebagai *public key*. Sedangkan dekripsinya dilakukan menggunakan kunci pasangannya yang disebut sebagai *private key/secret key*. Kita mempublikasikan *public key* kita ke semua orang, sehingga siapapun bisa mengenkrip data (menggunakan *public key* kita) dan mengirimkan kepada kita, sehingga hanya kita-lah yang bisa membaca data tersebut (menggunakan *secret key* kita). Tentu saja kita harus menjaga kerahasiaan *secret key* kita.

Pasangan public key dan private key didesain dengan algoritma sedemikian rupa sehingga (hampir) tidak mungkin untuk mencari fungsi *private key* hanya berdasarkan fungsi *public key* yang dipublikasikan tersebut. Siapapun yang memiliki *public key* bisa melakukan enkripsi data tetapi tidak bisa melakukan dekripsi. Hanya orang yang memiliki *secret key* pasangannya yang bisa mendekripsi data tersebut.

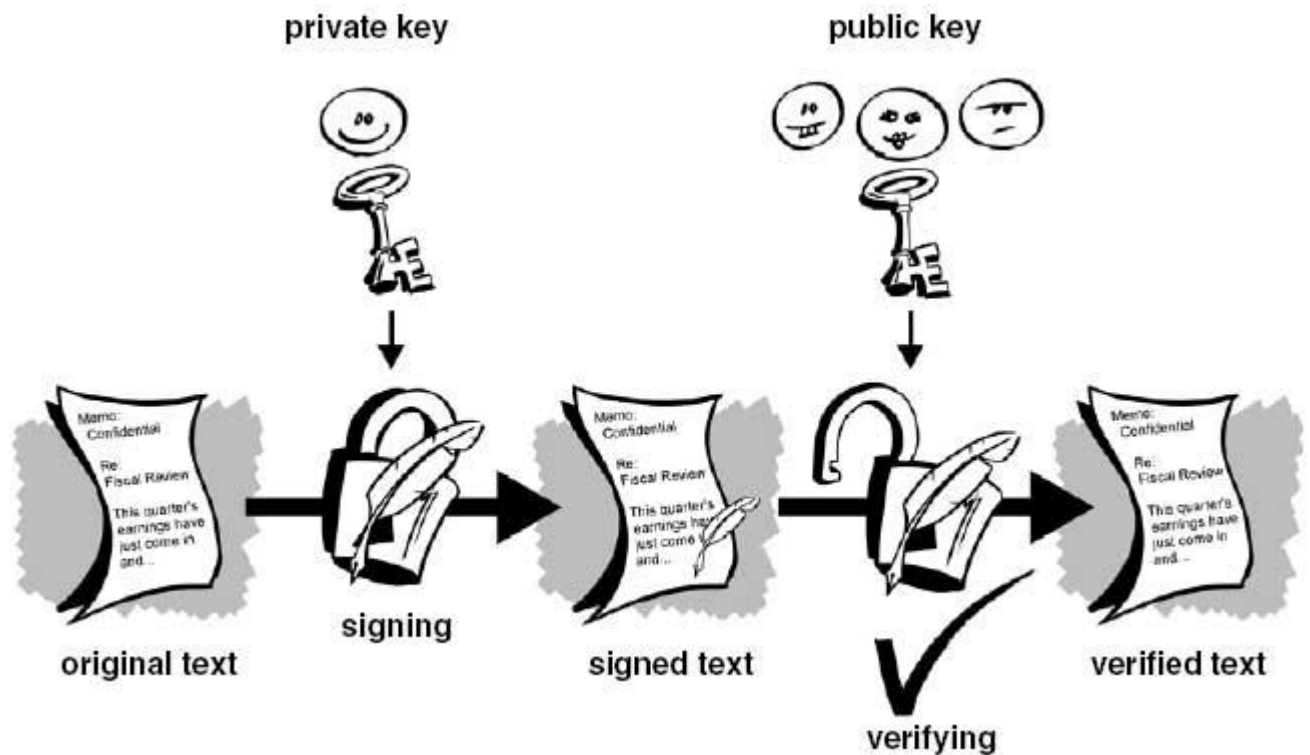


4. Digital Signature

Tanda tangan digital digunakan untuk memberi kepastian bahwa sebuah data benar-benar dikirimkan oleh pengirim yang sebenarnya. Jadi dengan kata lain memastikan tidak ada orang lain yang mangaku-ngaku sebagai kita dan mengirimkan data pada seseorang, padahal sesungguhnya kita tidak mengirim data tersebut.

Digital Signature menerapkan hal yang mirip dengan enkripsi asimetris, tetapi menggunakan mekanisme yang sebaliknya. Ketika kita mengirimkan sebuah data, maka kita membubuhkan tanda tangan digital kita yang diramu oleh *private key* kita.

Pada sisi penerima, si penerima akan memverifikasi tanda tangan yang kita sertakan tadi menggunakan *public key* kita yang sudah tersebar dan diketahui sebelumnya. Kalau verifikasi tanda tangan tadi cocok (*public key* match dengan *secret key* dalam tanda tangan), berarti bisa dipastikan bahwa kiriman data tadi otentik dari pengirim sebenarnya.



B. Implementasi Kriptografi pada Mail User Agent

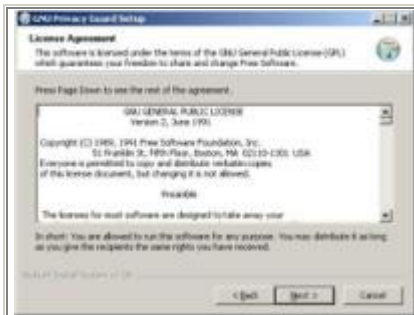
1. Instalasi GnuPG

Sebelum instalasi yang terkait dengan MUA, kita harus melakukan instalasi program GnuPG terlebih dahulu. Installer terbaru pada saat dokumen ini ditulis adalah gnuPG-w32cli-1.4.2.exe yang bisa didownload di [sini](#). Step-step instalasi bisa dilihat pada screenshot berikut:

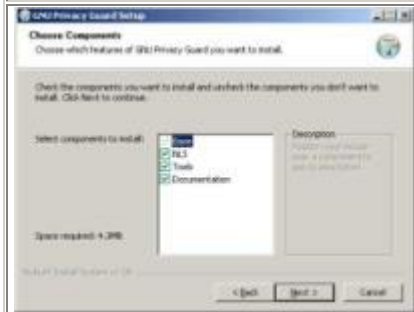


Pilih bahasa yang anda inginkan untuk instalasi

Klik Next



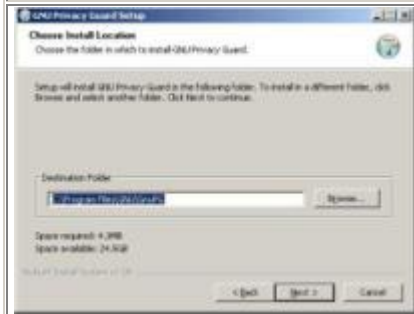
Read them carefully



Pilih komponennya (default is suggested)



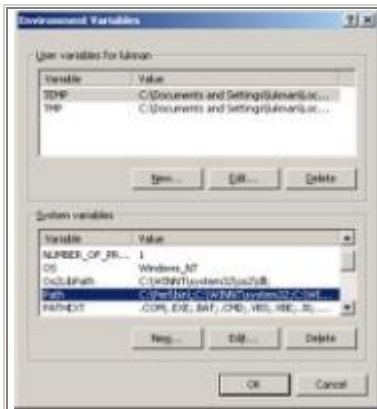
Pilih bahasa lagi



Pilih lokasi Instalasi



Pilih icon group



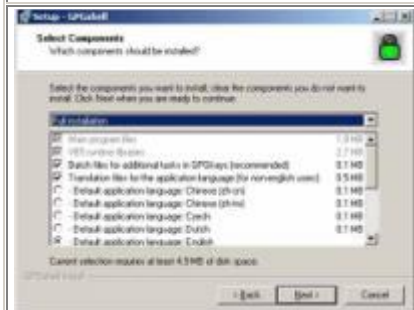
Pada box di bawah, pilih **Path** | kemudian klik **Edit**

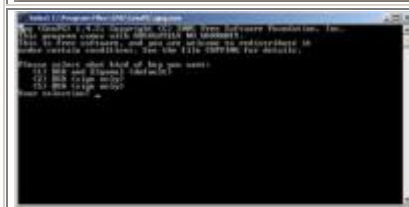
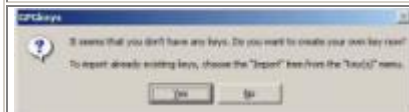
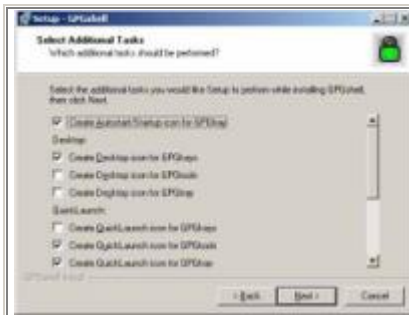


Pada **Variable Value** | tambahkan C:\Program Files\GNU\GnuPG dan untaian sebelumnya beri tanda ; (titik koma)

2. Instalasi GPGShell

Installer terbaru adalah `pggsh346.zip` yang bisa didownload di [sini](#).





Akan muncul beberapa pertanyaan via console dan harus kita jawab. Contoh pertanyaan dan jawabannya adalah sebagai berikut:

Please select what kind of key you want:

- (1) DSA and Elgamal (default)
- (2) DSA (sign only)
- (5) RSA (sign only)

Your selection?

===== **Tekan enter untuk pilah default**

DSA keypair will have 1024 bits.
ELG-E keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)

===== **Tekan enter untuk pilah default**

Please specify how long the key should be valid.
0 = key does not expire
<n> = key expires in n days
<n>w = key expires in n weeks
<n>m = key expires in n months
<n>y = key expires in n years
Key is valid for? (0)

===== **Tekan enter untuk pilah default**

Key does not expire at all
Is this correct? (y/N) y

===== **Tekan enter untuk pilah default**

You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:
"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Lukman HDP
Email address: lukman@bpk.go.id
Comment: Sys Admin
You selected this USER-ID:
"Lukman HDP (Sys Admin) <lukman@bpk.go.id>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o

===== **Sudah jelas dari tulisan di atas, to??**

You need a Passphrase to protect your secret key.

Repeat passphrase:

===== **Masukkan passphrase. Ini adalah semacam password yang digunakan untuk mengakses key kita nanti.**

We need to generate a lot of random bytes. It is a good idea to perform some other action (type on the keyboard, move the mouse, utilize the disks) during the prime generation; this gives the random number generator a better chance to gain enough entropy.

.+++++
+++++



Maka email akan mendapatkan tambahan header dan footer sebagai digital signature dengan bentuk ketika email belum mendapatkan tanda tangan digital tsb.

Kemudian kirim seperti biasa.

4. Mengirim email dengan menggunakan enkripsi

Seerti disebutkan diatas, bahwa untuk mengirim email terenkripsi kita harus memiliki *public key* milik orang yang akan menjadi tujuan email tsb. Jadi kita harus melakukan tukar menukar *public key* dengan orang lain. cara yang paling sederhana adalah dengan mengeksport *public key* kita ke dalam sebuah file, kemudian mendistribusikannya baik secara online maupun offline.

Cara mengeksport *public key* adalah sebagai berikut:



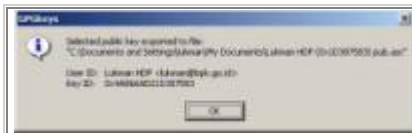
Masuk ke **Start | Program | GPGShell | GPGTools**



Klik kanan pada key atas nama kita, kemudian pilih export



Kita akan diminta menentukan lokasi dimana akan menyimpan file export ke sesuka hati



Akan muncul jendela konfirmasi



Selanjutnya ada pertanyaan apakah kita juga akan mengekspor secret key kita doing, don't you? heheh

Tentu saja saja kita tidak akan mengekspor secret key dan membagikannya

Sesudah terekspor, kita bisa bertukar public key dengan orang lain. Apabila kita sudah mendapatkan file .asc milik teman kita, maka kita harus mengimportnya ke dalam database key kita.

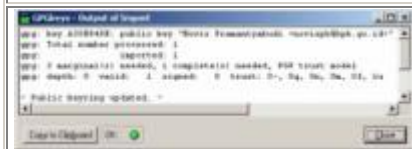
Caranya:



Pada jendela GPGkeys, klik menu **Key(s) | Import**



Kemudian kita akan diminta menentukan lokasi file .asc yang kita dapat dari teman dengan di mana anda menyimpannya.



Jika sukses, maka akan muncul konfirmasi bahwa key telah terimport



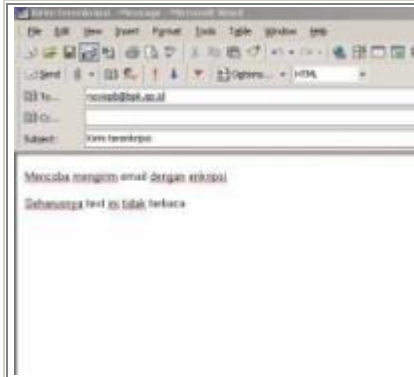
Maka di daftar kan muncul entry yang baru kita import

Mungkin pada waktu import akan muncul dialog sebagai berikut:

It is NOT certain that the key belongs to the person named in the user ID. If you *really* know what you are doing, you may answer the next question with yes.

Use this key anyway? (y/N) y

===== Jawab (y)es



Tulis email seperti biasa



Kemudian klik kanan pada icon GPGtray, pilih **Current Window | Encrypt**



Kemudian kita diminta memilih key yang akan kita gunakan untuk mengenkripsi, disesuaikan dengan siapa orang yang akan kita kirim email.



Maka text yang tadinya clear akan menjadi tidak bisa dibaca

Demikianlah proses mengirim email terenkripsi menggunakan GPGShell dan Outlook

5. Mengirim email terenkripsi menggunakan Mozilla Thunderbird dan Enigmail

Coming soon... :)

Penutup

Demikian, tulisan singkat ini. Struktur tulisannya pasti kacau ya :) *feedback*-nya ya

Change Log

2 Oktober 2005

- Penulisan pertama dokumen ini

24 Oktober 2004

- Finishing tulisan ini (lama yak ..)

Referensi

1. Google.com
2. Intro To Crypto.pdf
3. Beberapa sumber yang lain dari internet

Thanks To

1. My Dad and Mom, My Sisters.
 2. PDE@BPK crews.
 3. Happy-tos dan A-Mild.
 4. Semua yang telah membantu penulisan dokumen ini.
-