Studi Mengenai Secure Email Dengan Aplikasi PEM dan PGP

Dewangga Respati – NIM : 13503120

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13120@students.if.itb.ac.id

Abstrak

Pada masa modern ini, penggunaan internet semakin meluas ke berbagai lapisan masyarakat. Penggunaan internet tersebut salah satunya adalah sebagai sarana komunikasi dalam bentuk *electronic mail (email).* Pesan yang disampaikan lewat *email* tersebut kadangkala merupakan pesan yang sangat penting dan rahasia sehingga harus dijaga kerahasiaannya dari pihak yang tidak berwenang. Dalam upaya menjaga kerahasiaan itulah dibuat semacam *secure-email* yang berupa suatu platform yang disebut S/MIME. S/MIME merupakan singkatan dari Secure Multipurpose Internet Mail Extensions.

Pada awal abad 90 dua aplikasi secure email muncul yaitu :

1. PEM : Private Enhanced Mail
2. PGP : Pretty Good Privacy

Dan dari dua aplikasi tersebut, PGP merupakan aplikasi yang menjadi standart untuk klien *secure email.* Oleh karena itu, penulis mempunyai ide untuk membahasnya dalam makalah yang akan dibuat. Makalah ini akan mempunyai judul *"Studi Mengenai Secure Email Dengan Aplikasi PEM dan PGP"*.

PGP adalah aplikasi kriptografi *high-security* yang memungkinkan orang untuk bertukar pesan atau file dengan tetap menjaga kerahasiaan atau *privacy*, melakukan *authentication*, dan juga berkomunikasi dengan nyaman. PGP dapat digunakan untuk mengenkripsi dan memberi *digital signature* pada file ataupun email. PGP dikembangkan pertama kali oleh Phil Zimmerman pada pertengahan tahun 80an. Versi pertama dirilis pada tahun 1991 melalui media internet. Dan perilisan ini mendapat perhatian dari NSA(National Security Agent).

Ada 3 isu utama sehingga menjadi pertimbangan digunakannya PGP dalam komunikasi melalui internet. Isu-isu tersebut antara lain :

1. Privacy
   Menyimpan dan mengalirkan data sehingga hanya orang-orang yang berhak yang dapat melihat isi data.
2. Integrity
   Memastikan bahwa isi file, data, dan aplikasi tidak dimodifikasi dengan tanpa izin dari pengirim pesan.
3. Authentication
   Memastikan bahwa pesan yang dikirim merupakan pesan asli dari pengirim awal, sehingga orang lain tidak dapat mengakui sebagai pengirim pesan dan juga pengirim pesan tidak dapat membantah bahwa dia yang mengirim pesan.

Kata Kunci: *Internet, email, S/MIME, PEM, PGP, secure email, high-security, privacy, authentication, digital signature*

1. Pendahuluan

Seperti yang telah disebutkan pada bagian abstrak, Pretty Good Privacy dikembangkan oleh Phillip Zimmerman untuk menyediakan komunikasi yang aman dalam lingkunagn komunikasi elektronik. "Pretty Good" dimaksudkan untuk menunjukkan bahwa PGP membutuhkan usaha keras bagi kriptanalis untuk memecahkan kode *cipherteks.* Hal ini lebih disebabkan karena framework prosedur enkripsi yang berbasis PKI (Public Key Infrastructure) dan standar enkripsinya (menggunakan Diffie-Helman atau RSA algorithm). PGP berkembang menjadi aplikasi yang tangguh dalam arahan pemiliknya yang sekarang yaitu Network Associates. Sampai versi PGP yang paling terakhir dirilis adalah total open source, sehingga memberi kewenangan pada orang-orang untuk

membaca kodenya dan memberi tanggapan atau saran.

Sekarang kita bertanya-tanya bagaimana PGP bekerja. Prinsip kerja PGP adalah sebagai berikut; Ketika seseorang mulai menggunakan PGP, PGP akan mulai *generate Key Pair.* Ini hanya merupakan file teks yang hanya seperti omong kosong kepada manusia. Kunci yang di-*generate* dapat dalam beberapa level kekuatan – 512, 1024, ataupun 2048 bit. Semakin tinggi jumlah bit yang digunakan kunci, maka semakin kuat nilai enkripsi dari kunci. Salah satu kunci dari pasangan kunci yang dihasilkan tadi adalah kunci privat – kunci ini harus selalu disimpan dengan aman dan jangan diberikan pada orang lain. Sedangkan kunci yang lain adalah kunci publik – kunci ini sebaiknya diberikan kepada banyak orang yang memungkinkan.

Aspek dari PGP yang seringkali digunakan adalah proses *signing* dan enkripsi dari email atau file. "Signing" dokumen adalah suatu cara untuk melakukan verifikasi integritas dari hasil kerja yang asli. Metode yang digunakan adalah sebagai berikut :

1. Membuat intisari atau "hash" dari file atau email. Hash adalah algoritma yang menghasilkan output (hash) yang unik dari input (pesan) yang diberikan.
2. Menambahkan hash yang dihasilkan pada akhir dari pesan.
3. Ketika seseorang ingin melakukan verifikasi untuk membuktikan bahwa pesan tidak dimodifikasi, maka penerima akan menjalankan algoritma hash pada pesan dan membandingkan hasil tersebut pada hash yang ada pada akhir dari pesan. Jika signature yang dibandingkan cocok, maka dapat ditarik kesimpulan bahwa pesan tersebut belum pernah diubah.

Cara di atas dapat dianalogikan dengan contoh di bawah ini:
Hash Algorithm : Ambil setiap tiga huruf dari pesan (abaikan tanda baca) dan konversi huruf ke angka (a=1, b=2, ... , z=26). Gabungkan angka-angka tersebut.
Pesan :
Hello, This is a sample message to demonstrate signatures.
Proses Hash Algorithm :
He**l**lo, **T**hi**s** is **a** sa**m**p**l**e me**s**sa**g**e t**o** de**m**on**s**tr**a**te **s**ig**n**at**u**re**s.**
12 +20 +19 +1 +13 +5 +19 +7 +15 +13 +19 +1 +19 +14 +21 +19
= 217 (oleh karena itu hash value adalah 217)
Pesan setelah ditambahkan hash value menjadi:
Hello, This is a sample message to demonstrate signatures.
Hash value: 360
Jika pesannya telah diubah, maka nilai hash value tidak akan sama.
Altered message:
Hello, This is an altered message to demonstrate signatures.
Membuat hash baru:
He**l**lo, **T**hi**s** is **a**n a**l**t**e**red **m**e**s**sa**g**e to dem**o**ns**t**ra**t**e s**i**gn**a**tu**r**es.
12 +20 +19 +1 +12 +18 +13 +19 +5 +4 +15 +20 +20 +9 +1 +18
= 206 (oleh karena itu hash value adalah 206)
Karena hash masing-masing pesan tidak sama, maka dapat disimpulkan bahwa pesan telah diubah.
Hashing algorithm yang sebenarnya jauh lebih kompleks. Sebagai tambahan, hashing algorithm digunakan bersamaan dengan kunci privat untuk menghasilkan signature yang unik. Oleh karena itu jika orang yang berbeda memberi tanda pada Email yang sama, signature yang dihasilkan akan berbeda. Lalu kunci publik dari pasangan kunci digunakan untuk membandingkan hash yang dihasilkan oleh kunci privat, dan jika hash cocok, maka dua hal dapat dijamin :

1. Pesan tidak pernah dimodifikasi sejak diberi tanda.
2. Signature tidak pernah dipalsukan.

Enkripsi adalah metode untuk mengubah plainteks (teks yang dapat dibaca oleh manusia artinya) menjadi cipherteks (teks yang tidak bermakna). Ada banyak cara untuk melakukan enkripsi, beberapanya lebih kuat daripada yang lainnya. Dua kategori utama dari enkripsi adalah simetrik dan asimetrik. Dalam kriptografi simetrik, kunci untuk enkripsi sama dengan kunci untuk dekripsi. Dalam kriptografi asimetrik dimana PGP termasuk di dalam kategori ini, satu kunci untuk enkripsi (*public key*) dan kunci yang lain untuk dekripsi (*private key*). Jadi jika pengguna A ingin mengirim pesan terenkripsi ke pengguna B, pengguna A pertama kalinya harus mengambil *public key* dari pengguna B. Hal ini memungkinkan karena *public key* memang untuk disebarkan. Lalu pengguna A melakukan enkripsi pesan menggunakan *public key* dari pengguna B. Pesan yang terenkripsi sekarang hanya dapat didekripsi dengan menggunakan *private key* dari B yang

hanya dimiliki oleh pengguna B. Bahkan pengguna A yang menuliskan pesan tidak dapat melakukan proses dekripsi karena dia tidak memiliki kunci privat B. Hal ini memastikan bahwa pesan tidak dapat dimengerti oleh orang lain selain pengguna A. Enkripsi dan proses *signing* biasanya dikombinasikan. Dalam skenario seperti ini, pengguna A akan menggunakan *public key* B untuk mengenkripsi pesan, lalu menggunakan *private key* dia untuk memberi *sign* ke pesan. Hal ini memastikan bahwa tidak ada siapapun selain pengguna B yang dapat membaca pesan. Dan ketika pengguna B menerima pesan, maka dapat dipastikan bahwa pesan belum pernah diubah. Untuk membaca pesan, pengguna B pertama menggunakan *public key* dari A untuk melakukan verifikasi bahwa *signature* nya cocok. Lalu pengguna B akan menggunakan *private key* dia untuk melakukan dekripsi pesan yang ditulis oleh A.

2. Servis yang Dilakukan PGP

Operasi yang dilakukan oleh PGP terdiri dari lima servis umum, antara lain :

1. Authentication
   Sender Authentication memiliki arti bahwa pengirim memberikan tambahan berupa *digital signature* ke emailnya dan penerima pesan akan melakukan verifikasi menggunakan *public key cryptography.* Berikut ini merupakan contoh dari operasi *authentication* yang dilakukan antara pengirim dan penerima pesan :
   a. Pada sisi pengirim, fungsi hash SHA-1 digunakan untuk membuat 160 bit pesan dari pesan email yang keluar.
   b. Pesan tadi dienkripsi dengan RSA menggunakan *private key* dan hasilnya dihubungkan ke pesan. Pesan yang merupakan gabungan dikirimkan ke penerima.
   c. Penerima pesan menggunakan RSA dengan *public key.*
   d. Penerima pesan membandingkan *message digest* yang dihitung secara lokal dengan *message digest* yang diterima.

   Deskripsi di atas menggunakan RSA/SHA digital signature. PGP juga mendukung DSS/SHA signature. DSS merupakan singkatan dari Digital Signature Standart. Deskripsi di atas juga berbasiskan pada menempelkan *signature* ke dalam pesan. PGP juga mendukung *detached signature* yang dapat dikirimkan terpisah ke penerima pesan. *Detached Signature* sangat berguna jika dokumen harus ditandai oleh lebih dari satu orang.

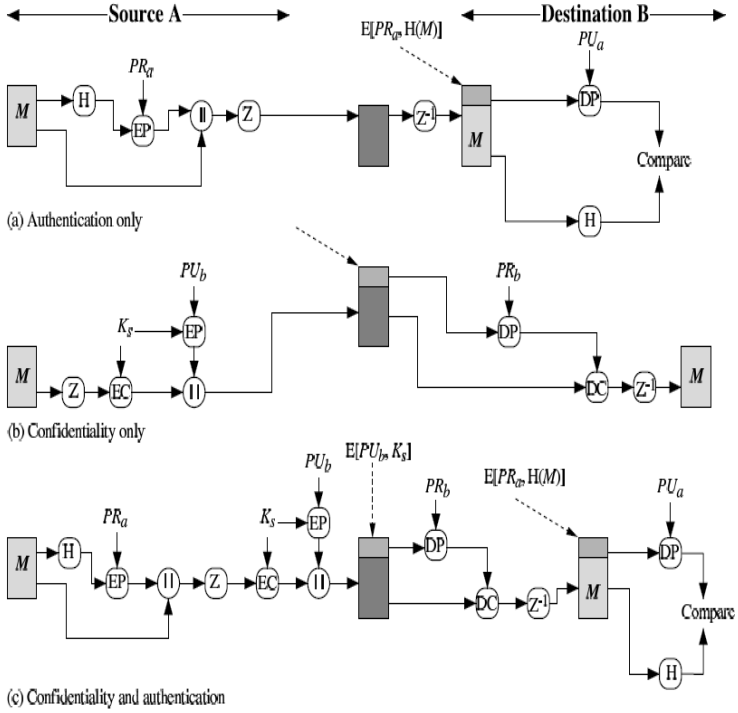
2. Confidentiality
   Servis in dapat pula digunakan untuk melakukan enkripsi file disk. PGP menggunakan *symmetric-key encryption* untuk servis ini. Pengguna memiliki tiga pilihan *block cipher algorithm* yang berbeda untuk servis ini: CAST-128, IDEA, atau 3DES, dengan CAST-128 sebagai pilihan default.
   - Block Cipher digunakan dalam CFB (Cipher Feedback) Mode.
   - Kunci enkripsi 128 bit, yang disebut *session key*, dihasilkan untuk setiap pesan email secara terpisah.
   - *Session key* dienkripsi dengan menggunakan RSA dengan *public key* dari penerima. Alternatif lain, *session key* dapat pula dienkripsi dengan menggunakan ElGamal algorithm. ElGamal adalah variasi dari Diffie-Hellman yang memungkinkan untuk enkripsi dan dekripsi.
   - Apa yang dikirimkan lewat jaringan adalah pesan email setelah dienkripsi dengan menggunakan *session key* dan *session key* setelah dienkripsi dengan menggunakan *public key* dari penerima.
   - Jika *confidentiality* dan *sender-authentication* diperlukan secara simultan, *digital signature* untuk pesan dihasilkan dengan menggunakan *hash code* dari pesan plainteks dan ditambahkan pada pesan email sebelum dienkripsi dengan menggunakan *session key.*

3. Compression
   PGP secara otomatis melakukan kompresi pesan email setelah menerapkan *signature* tapi sebelum melakukan enkripsi. Hal ini untuk memungkinkan penyimpanan jangka panjang dari pesan yang belum terkompres bersama dengan *signature* nya. Hal ini juga memisahkan algoritma enkripsi dari prosedur verifikasi pesan. Proses kompresi dilakukan dengan menggunakan algoritma ZIP.

4. E-mail compatibility
   Karena enkripsi, bahkan setelah dibatasi oleh *signature*, menghasilkan *binary string* yang berubah-ubah. Dan sejak sistem email yang hanya membolehkan penggunaan karakter ASCII, kita harus dapat merepresentasikan *binary data* dengan *string* ASCII. PGP menggunakan *radix 64 encoding* untuk tujuan ini.

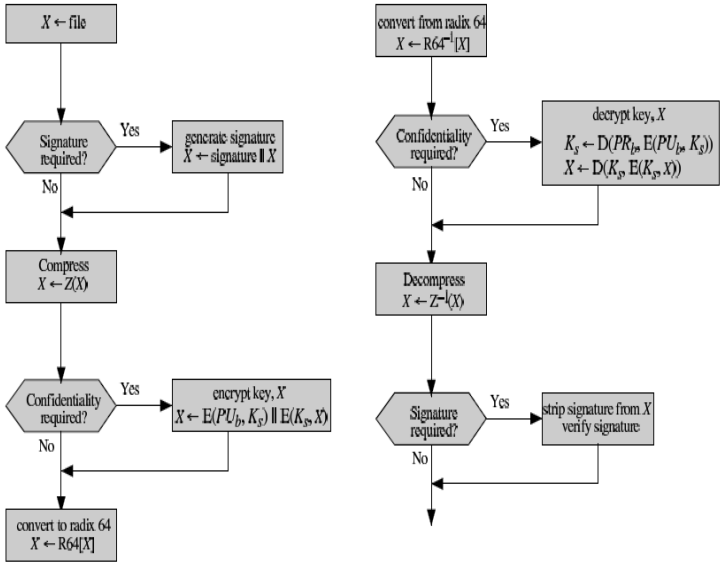
5. Segmentation
   Untuk pesan email yang panjang, banyak sistem email yang memberi batasan pada berapa banyak pesan yang akan dikirimkan sebagai unit. Beberapa sistem email memecah pesan email menjadi 50.000 byte segment dan mengirimkan segmen-segmen tersebut secara terpisah. PGP memiliki fasilitas yang terintegrasi untuk melakukan segmentasi dan penggabungan pesan kembali.

Gambar di samping menunjukkan tiga model yang berbeda dimana PGP dapat digunakan untuk mekanisme *secure email*. Diagram paling atas digunakan jika hanya menggunakan servis *auhentication*. Diagram tengah menunjukkan penggunaan *confidentiality*, dan gambar terakhir jika keduanya baik *authentication* maupun *confidentiality* dibutuhkan. Simbol EP dan DP dalam gambar menunjukkan *public-key* enkripsi dan dekripsi; EC dan DC menunjukkan *symmetric-key* enkripsi dan dekripsi; simbol Ks untuk *session-key* untuk *symmetric-key* enkripsi/dekripsi. H untuk hashing; jj untuk penggabungan; Z untuk kompresi menggunakan algoritma ZIP, dan R64 untuk konversi ke radix 64 ASCII format.



Gambar 1. PGP Crypthographic Function

Sedangkan gambar di bawah ini menunjukkan proses pengiriman dan penerimaan pesan dari *sender* ke *receive.*



Gambar 2. Pengiriman dan Penerimaan Pesan PGP

3. Manajemen Kunci dalam PGP

Seperti yang telah kita lihat, *public key encryption* adalah inti dari PGP. Hal ini digunakan untuk dua tujuan: pengirim menggunakan kunci privatnya untuk

menggantikan *digital signature* nya dalam pesan yang keluar, dan pengirim menggunakan *public key* dari penerima untuk melakukan enkripsi *session key* yang rahasia.

Kita dapat mengharapkan agar suatu orang memiliki banyak kunci publik (yang tentu saja berhubungan dengan kunci privat). Ini memungkinkan karena seseorang dalam proses penerimaan kunci publik yang lama, tetapi diperbolehkan dalam periode transisi, memutuskan untuk membuat dua kunci baik yang baru ataupun yang lama menjadi *available* untuk saat itu. Beberapa orang juga mungkin memilih *multiple public key* untuk alasan-alasan yang lain.

Jadi PGP harus memberi wewenang untuk kemungkinan jika penerima pesan mempunyai *multiple public keys.* Hal ini memunculkan beberapa pertanyaan :

- Mari kita lihat jika PGP menggunakan satu dari kunci publik bagi penerima, maka bagaimana cara penerima mengetahui kunci publik mana yang digunakan.
- Jika pengirim menggunakan salah satu dari *multiple private key* yang pengirim punya untuk menandai pesan, maka bagaimana penerima mengetahui *public key* mana yang berkoresponden.

PGP mengatasi masalah ini dengan menggunakan penerapan *key identifiers* yang cukup pendek (key ID) dan menerapkan untuk setiap agen PGP memiliki daftar *private/public keys* dari agen itu sendiri, bersama dengan *key ID* nya, dan daftar kunci publik bersama dengan *key ID* yang berasosiasi, untuk semua email yang berkoresponden. Daftar pertama dikenali sebagai *private key ring* dan yang terakhir adalah *public key right*. Contoh dari *private key ring* dan *public key ring* dapat dilihat pada gambar di halaman selanjutnya. Kunci untuk user tertentu adalah unik yaitu merupakan kombinasi dari *user ID* dan *key ID. Key ID* berasosiasi dengan *public key* yang terdiri dari paling minimal 64 bit.

Kita kembali ke *private key ring* yang ditunjukkan pada gambar di halaman selanjutnya, untuk alasan keamanan, PGP menyimpan kunci prvat dalam tabel dalam format yang terenkripsi sehingga kunci hanya dapat diakses oleh pengguna yang memilikinya. PGP dapat menggunakan semua algoritma block cipher yang ada yaitu CAST-128, IDEA, dan 3DES dengan CAST-128 digunakan sebagai pilihan *default* untuk algoritma enkripsi. Algoritma enkripsi meminta pengguna untuk memasukkan *pass-phrase.* Pada *pass-phrase* diterapkan fungsi hash dengan SHA-1 untuk mendapatkan 160-bit kode hash. 128 bit yang pertama dari *hash code* digunakan untuk kunci enkripsi dari algoritma CAST-128. Keduanya baik *pass-phrase* dan *hash code* akan dengan segera dibuang.

Dengan melihat kepada *public key ring* yang ditunjukkan pada gambar di halaman selanjutnya, *Fields Owner Trust, Key Legitimacy, Signature*, dan *Signature Trust* adalah untuk memutuskan seberapa percayakah kita untuk memberikan kunci publik pada orang lain [Jika A mempunyai kunci publik dari B dalam *ring*, tapi kunci tersebut sebenarnya dimiliki oleh C, maka C dapat mengirim pesan ke A dan memalsukan signature dari B (A akan berpikir bahwa pesan yang dikirim oleh B merupakan pesan yang dikirim oleh C) dan semua pesan terenkripsi dari A ke B akan dapat terbaca oleh C]. Nilai dari kunci *legitimacy field column* dihitung oleh PGP. Nilai ini memberitahukan PGP tentang parameter kepercayaan untuk meletakkan kunci publik dalam hubungannya dengan baris koresponden untuk menjadi kunci yang valid untuk *user ID.*

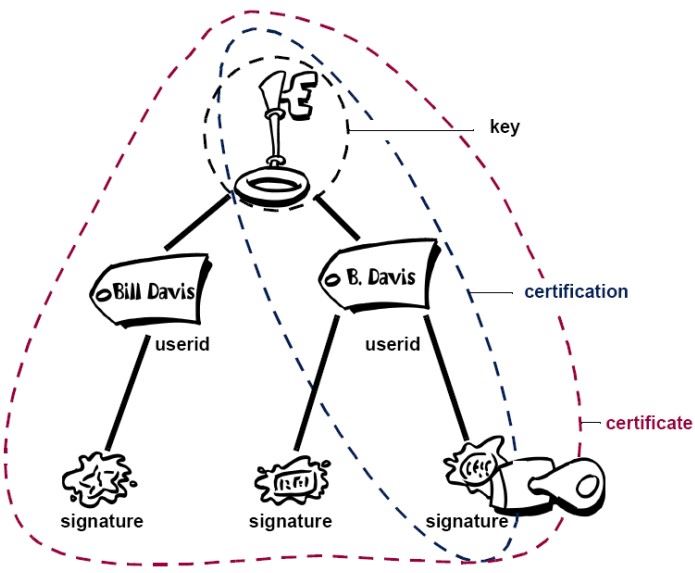


Figure 1-8. Anatomy of a PGP certificate

Masukan yang disimpan pada *public key field* sebenarnya adalah suatu sertifikat. *Signature field* mengandung *signature* dari satu atau lebih otoritas sertifikat yang sudah disahkan. Setiap

signature mempunyai asosiasi dengan suatu nilai dari *signature field* yang mengindikasikan seberapa banyak kepercayaan PGP terhadap pengesah dari sertifikat yang diterima. Nilai dari kunci *legitimacy field* diturunkan dari nilai yang disimpan dalam *signature trust field.* Masukan dalam *trust field* pemilik di tabel *public key-ring* mengindikasikan perluasan untuk pemilik dari *public key* mana yang dapat dipercaya untuk memberi tanda pada sertifikat. Nilai ini diberikan oleh pengguna untuk siapa *public-key-ring* ini ditujukan. Gambar selanjutnya menunjukkan bagaimana format umum dari pesan PGP. Seperti yang ada pada gambar, pesan PGP terdiri dari 3 komponen : komponen *session key, signature,* dan pesan email itu sendiri. Diharapkan hanya masukan yang tidak disangka adalah dua byte pertama dari pesan. Ini dilakukan untuk memungkinkan bagi penerima pesan untuk menyimpulkan bahwa kunci publik yang benar (dari pengirim) telah digunakan untuk melakukan dekripsi pada pesan untuk proses *authentication.* Dua oktet ini berlaku juga sebagai 16-bit frame untuk melakukan *sequence check* terhadap pesan email yang sebenarnya. Pesan itu selanjutnya dihitung kembali menggunakan SHA-1.

Private Key Ring

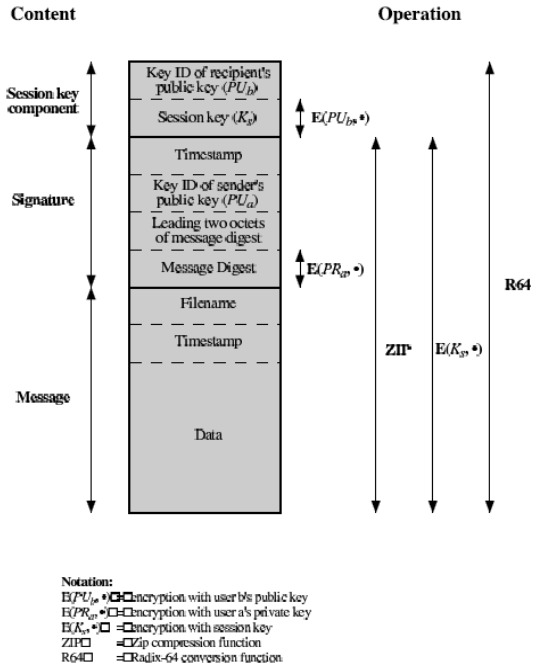
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
T_i	$PU_i \bmod 2^{64}$	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

* = field used to index table

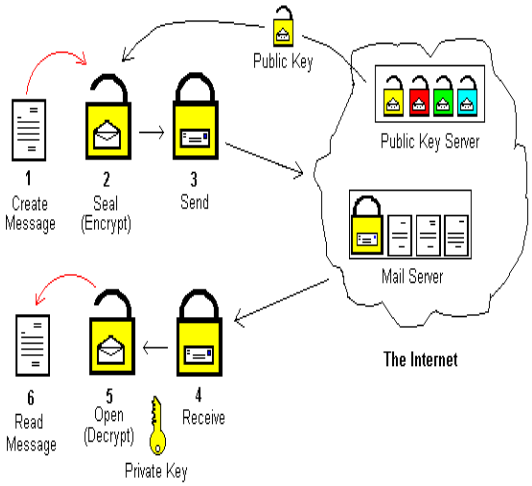
Gambar 3. Struktur Umum dari Private dan Public Key Ring



Gambar 4. Format Umum dari Pesan PGP

4. Algoritma Simetris Dalam PGP

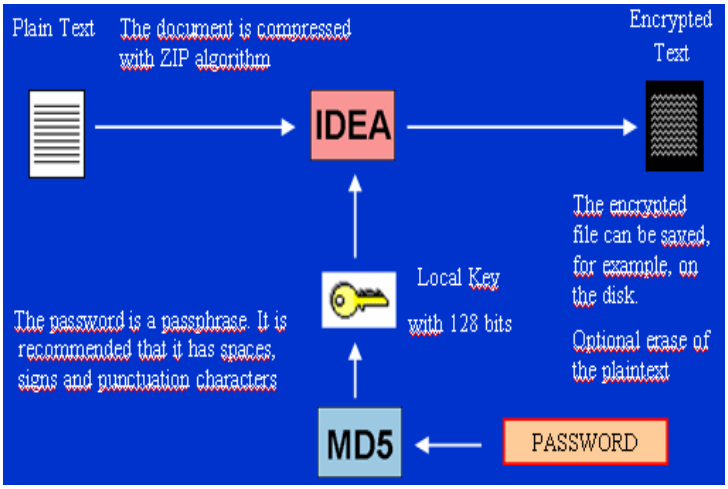
PGP menawarkan seleksi dari algoritma kunci rahasia yang berbeda untuk melakukan enkripsi terhadap pesan yang sebenarnya. Dengan algoritma kunci rahasia, kita mengartikan bahwa konvensional atau *symmetric block cipher* yang menggunakan kunci yang sama untuk enkripsi dan dekripsi. Tiga *symmetric block cipher* yang ditawarkan oleh PGP adalah CAST, Triple-DES, dan IDEA. Ketiganya bukanlah algoritma yang sederhana, melainkan ditemukan oleh tim kriptografer dengan reputasi yang istimewa. Ketiga cipher tersebut beroperasi dalam 64-bit block dari plainteks dan cipherteks. CAST dan IDEA mempunyai panjang kunci 128 bit, sedangkan Triple-DES menggunakan 168-bit. Seperti pada Data Encryption Standard (DES), semua dari cipher ini dapat digunakan dalam Cipher Feedback (CFB) ataupun Cipher Block Chaining modes. PGP menggunakan mereka dalam 64-bit CFB mode. Gambar di bawah ini menunjukkan cara kerja dari PGP.



Phil Zimmerman memasukkan algoritma enkripsi CAST dalam PGP karena algoritma ini menjanjikan suatu block cipher yang bagus dengan ukuran kunci 128-bit, algoritma ini juga sangat cepat dan juga gratis. Nama CAST sendiri didapat dari nama inisial dari perancang algoritma ini, Carlisle Adams and Stafford Tavares of Northern Telecom (Nortel).

Nortel memeiliki hak paten terhadap algoritma CAST ini, tetapi mereka telah membuat komitmen dalam kesepakatan untuk membuat CASt ini dimungkinkan untuk diakses oleh semua orang secara cuma-cuma. CAST memiliki desain yang sangat luar biasa, oleh orang dengan reputasi yang bagus di bidang kriptografi ini. Desainnya berbasis pada pendekatan yang sangat formal, dengan beberapa pernyataan formal yang dapat dibuktikan yang memberikan alasan yang bagus untuk mempercayai bahwa CAST ini membutuhkan *key exhaustion* untuk memecah kunci 128-bit nya. Ada argumen yang kuat yang membuktikan bahwa CAST benar-benar kebal terhadap linear maupun differential cryptanalysis, dua dari cryptanalysis yang paling kuat. Seperti yang disebutkan pada beberapa literatur kedua cara cryptanalysis ini sangatlah efektif untuk memecah Data Encryption Standard (DES). CAST merupakan algoritma yang terlalu baru untuk ditemukan *track record* nya, tapi CAST memiliki desain yang formal dan reputasi dari desainernya juga sangat bagus sehingga tanpa diragukan dapat menarik perhatian dan membuat cryptanalis untuk berusaha menyerang algoritma ini. Phil mengatakan bahwa dia memiliki keyakinan yang sama dari CASt ini seperti juga dengan keyakinan pada saat IDEA, cipher yang digunakan dalam versi terakhir dari PGP. Pada saat itu, IDEA juga terlalu baru untuk dibuat *track record* nya, tapi IDEA ini bertahan dengan cukup bagus. IDEA (International Data Encryption Algorithm) block cipher berdasarkan pada konsep desain dari mencampurkan beberapa operasi dari grup aljabar yang berbeda. IDEA ini ditemukan pada ETH di Zurich oleh James L.Massey dan Xuejia Lai, dan dikenalkan ke publik pada 1990. Pada awal-awal pengenalan ke publik algoritma ini dinamakan IPES (Improved Proposed Encryption Standard), tetapi akhirnya dirubah namanya menjadi IDEA.

Sejauh ini, IDEA dapat bertahan dari serangan jauh lebih baik daripada cipher yang lain seperti FEAL, REDOC-II, LOKI, Snefru dan Khafre. Dan IDEA lebih kuat juga daripada DES dalam menghadapi serangan cryptanalis, baik itu serangan dari linear maupun differential cryptanalysis. Gambar di samping menunjukkan skema dari enkripsi dengan menggunakan IDEA. Semakin lama IDEA ini semakin menarik perhatian dari kriptanalis untuk memecahkan algoritma in, tapi seiring dengan itu semakin berkembang pula algoritma dari IDEA ini. Sayangnya, halangan terbesar dari diterimanya IDEA ini sebagai standard adalah fakta bahwa Ascom Systec yang memegang paten dari IDEA ini. Dan tidak seperti DES dan CAST, IDEA ini tidak diberikan secara cuma-cuma kepada orang-orang lain.



PGP mengandung tiga kunci Triple-DES dalam daftar-daftar dari block cipher yang memungkinkan. DES ditemukan oleh IBM dalam pertengahan 1970 an. Meskipun mempunyai desain yang bagus, 56-bit kuncinya itu terlalu kecil untuk standart kriptografi sekarang. Triple-DES sangat kuat, dan telah

dipelajari dengan baik dalamn beberapa tahun terakhir, jadi ini mungkin lebih aman daripada algoritma yang baru seperti CAST dan IDEA. Triple-DES adalah DES yang diterapkan tiga kali terhadap blok data yang sama, menggunakan tiga kunci yang berbeda, kecuali bahwa operasi DES yang kedua berjalan kebalikan, yaitu dalam mode dekripsi. Triple-DES lebih lambat jika dibandingkan dengan CAST maupun IDEA, tetapi kecepatan biasanya bukan isu yang penting dalam aplikasi email. Meskipun Triple-DES menggunakan kunci dalam ukuran 168 bit, Triple-DES mempunyai kekuatan kunci dalam menghadapi penyerang yang menggunakan *data storage capacity* dalam strategi penyerangannya.

Berdasarkan pada *paper* yang diterbitkan oleh Michael Weiner dalam Crypto96, semua *data storage* yang masuk akal memungkinkan para penyerang untuk melakukan penyerangan mungkin membutuhkan usaha pemecahan setara dengan pemecahan kunci 129 bit. Berdasarkan *paper* yang diterbitkan, Triple-DES tidak dibebani oleh paten manapun. Kunci publik dari PGP yang digenerate oleh PGP versi 5 dan yang terakhir mempunyai informasi yang diembed di dalamnya yang memberitahukan pada pengirim block cipher apa yang dimengerti oleh aplikasi penerima, sehingga aplikasi penerima mengetahui cipher mana yang bisa digunakan untuk enkripsi. Kunci publik DSS menerima CAST, IDEA, atau Triple-DES sebagai block cipher, dengan CAST sebagai pilihan default. Saat ini, untuk suatu alasan, kunci RSA tidak menyediakan fitur ini. Hanya copher IDEA yang digunakan oleh PGP untuk mengirim pesan ke kunci RSA, karena versi PGP yang lebih lama hanya mendukung RSA dan IDEA.

2.2.1 Kompresi Data PGP

PGP normalnya melakukan kompresi pada plainteks sebelum melakukan enkripsi. Karena ini terlalu telat untuk melakukan kompresi pada plainteks jika setelah diterapkan enkripsi pada plainteksnya; data terenkripsi tidak memungkinkan untuk dikompres. Data compression menyimpan waktu transmisi dan disk space, dan yang lebih penting kemanan algoritma kriptografi yang lebih kuat. Banyak dari teknik kriptanalis memanfaatkan redundansi yang ditemukan dalam plainteks untuk memecahkan cipherteks. Kompresi data mengurangi redundansi dalam plainteks, sehingga memberikan rintangan yang cukup besar bagi kriptanalis. Hal ini membutuhkan waktu yang ekstra untuk melakukan kompresi pada plainteks, tapi dari sisi keamanan hal ini memiliki keuntungan yang berarti. File yang terlalu kecil untuk dilakukan kompresi, atau yang tidak terkompres dengan baik, tidak akan dikompres oleh PGP. Sebagai tambahan, program mengenali file yang diproduksi dengan menggunakan program kompresi yang populer seperti PKZIP, dan tidak melakukan kompresi pada file yang telah merupakan file hasil kompresi. Secara teknik, program ini menggunakan freeware kompresi ZIP yang dituliskan oleh Jean-Loup Gailly, Mark Adler, and Richard B.Wales. Program ZIP ini menggunakan algortima kompresi yang secara fungsional ekivalen dengan algortima yang digunakan pada PKWare's PKZIP 2.x. Program kompresi ZIP ini dipilih oleh PGP karena program ini memiliki rasio kompresi yang bagus dan juga karena program ini sangat cepat.

2.2.2 Angka Acak untuk Session Key

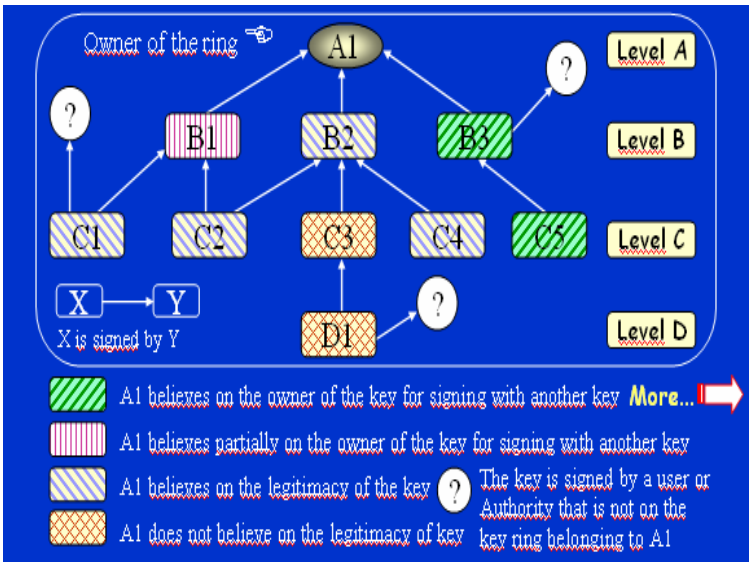
PGP menggunakan cryptographically strong pseudo-random-number generator untuk membuat temporary session key. Jika file random ini tidak tersedia, maka secara otomatis akan dibuat dan diisi dengan nomer acak yang diturunkan dari kejadian acak yang didapatkan oleh PGP program dari menghitung waktu dari keystroke dan juga pergerakan mouse. Generator ini menempatkan kembali seed file setiap kali file tersebut digunakan, dengan mencampur materi baru yang secara partial didapat dari sumber lain yang acak. Hal ini menggunakan algoritma enkripsi yang umum sebagai mesin untuk mengenerate nomer acak. File seed mengandung baik seed material yang acak dan material kunci yang acak untuk digunakan sebagai kunci dari mesin enkripsi yang umum untuk generator nomer acak. Seed file yang acak ini seharusnya disembunyikan dari penyingkapan, untuk mengurangi resiko dari penyerang mendapatkan session key yang sebelumnya maupun sesudahnya. Penyerang akan membutuhkan waktu yang banyak untuk mendapatkan sesuatu yang berguna dalam merekam random seed file ini, karena file telah "dicuci" secara kriptografi sebelum dan sesudah digunakan. Karena ini merupakan hal yang penting untuk hati-hati agar informasi ini tidak jatuh ke tangan yang salah. Jika memungkinkan, buatlah file hanya dapat dibaca oleh kita. Tetapi jika hal ini tidak memungkinkan, jangan membiarkan orang lain mengkopi isi dari disk anda.

2.2.3 Cara Menjaga Public Key

Dalam sistem kriptografi kunci publik, kita tidak perlu melakukan proteksi dari kunci publik dari pencurian. Faktanya, lebih baik jika kunci publik tersebut secara luas disebarkan. Tapi lebih penting untuk memproteksi kunci publik dari pihak yang tidak berwenang, untuk memastikan kunci publik benar-benar dimiliki oleh orang yang memang memiliki hak. Ini mungkin merupakan hal yang paling penting dalam sistem kriptografi kunci publik. Mari kita lihat pada kejadian fatal yang berpotensi untuk terjadi, lalu mendeskripsikan bagaimana untuk menghindarinya secara aman dengan PGP.

Misalkan kamu ingin mengirimkan pesan pribadi kepada Alice. Kamu meminta kunci publik dari Alice dari electronic bulletin board system (BBS). Kamu melakukan enkripsi pesan kamu ke Alice dengan kunci publik dan mengirimnya ke Alice melalui fasilitas email dari BBS. Sayangnya, tanpa diketahui oleh kamu ataupun Alice, seseorang lain yang bernama Charlie masuk ke BBS dan menggenerate kunci publik dia sendiri dengan menggunakan user ID dari Alice dimasukkan ke dalam pesan tersebut. Dia secara tersembunyi menggantikan kunci palsunya dengan di tempat dari publik kunci Alice yang sebenarnya. Kamu tanpa diketahui menggunakan kunci palsu ini yang dimiliki oleh Charlie di samping kunci publik Alice. Semua proses terlihat normal karena kunci palsu mempunyai user ID dari Alice. Sekarang Charlie dapat melakukan dekripsi pesan karena dia memiliki kunci privat yang cocok. Charlie juga mungkin untuk melakukan enkripsi kembali pesan yang telah didekripsi dengan menggunakan kunci publik dari Alice dan mengirim kembali ke Alice sehingga tidak ada yang mendeteksi adanya suatu kesalahan. Lebih jauh lagi, Charlie bahkan dapat membuat signatrure dari Alice dengan kunci privatnya karena semua orang akan menggunakan kunci palsu untuk memeriksa Alice's signature. Satu-satunya cara yang dapat digunakan untuk menghindari bencana di atas adalah dengan menjaga orang lain untuk mencuri kunci publik.

Jika kamu mendapatkan kunci publik dari Alice secara langsung melalui Alice, hal itu tidak menjadi masalah. Tetapi hal itu menjadi rumit jika Alice berada jauh dari kita. Diharapkan kamu bisa mendapatkan kunci publik Alice dari teman kepercayaannya, David, seseorang yang mempunyai kopi dari kunci publik Alice. David dapat menandai kunci publik Alice, menjamin integritas dari kunci publik Alice. David membuat signature ini dengan kunci privat yang dimilikinya. Hal ini akan membuat tanda pada sertifikat kunci publik, dan akan menunjukkan bahwa kunci Alice belum pernah dipalsukan. Hal ini membutuhkan pengetahuan yang baik tentang kunci publim David untuk mengecek signature. Diharapkan David dapat menyediakan kopi dari kunci publik kita kepada Alice juga. David seterusnya akan berlaku sebagai "introducer" antara kita dan Alice. Kunci publik yang telah ditandai atas nama Alice dapat diupload oleh David ataupun Alice ke BBS, dan kamu dapat mendownloadnya terakhir-terakhir.



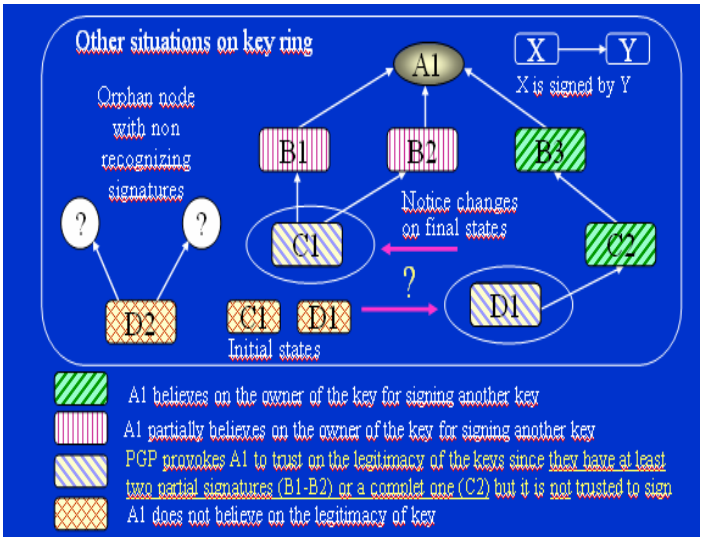
Selanjutnya, kamu dapat mengecek signature melalui kunci publik dari David dan seterusnya meyakinkan bahwa kunci yang digunakan adalah benar-benar kunci publik dari Alice. Tidak ada orang lain yang dapat membodohi kamu untuk menerima kunci palsu yang diterbitkannya sebagai kunci Alice karena tidak ada satu orangpun dapat megakali signature yang dikeluarkan oleh David. Pada intinya orang terpercaya dapat dimanfaatkan untuk menjalankan servis sebagai "introducer" antar pengguna dengan menyediakan signature bagi sertifikat kunci publiknya. Orang terpercaya ini dapat disebut sebagai "Certification Authority".

Semua kunci publik yang berkaitan dengan signature dari Certification Authority's dapat dipercaya selama kunci tersebut memang berkoresponden dengan orang yang berhak. Semua pengguna yang ingin berpartisipasi akan membutuhkan kopi yang bagus dari kunci publik Certification Authority, sehingga signature dari Certification Authority dapat diverifikasi. Dalam

beberapa kasus, Certification Authority mungkin juga berperan sebagai key server, membolehkan pengguna dalam jaringan untuk mencari kunci publik dengan menanyakan pada key server, tapi tidak ada alasan mengapa key server juga harus memastikan kunci. Certification Authority yang tersentralisasi biasanya sangat cocok untuk digunakan corporation yang terkontrol secara sentral maupun juga untuk institusi pemerintahan. Beberapa lingkungan institusi menggunakan hirarki dari Certification Authorities.

Untuk lingkungan yang lebih tidak sentralisasi, membolehkan semua user untuk beraksi sebagai introducer bagi temannya dan akan bekerja dengan lebih baik jika dibandingkan dengan centralized key certification authority. Salah satu fitur PGP adalah bahwa PGP dapat beroperasi dengan baik pada lingkungan yang tersentral dengan Certification Authority atau dalam lingkungan yang tidak sentral dimana setiap individu bertukar kunci pribadinya. Proses dalam menjaga kunci publik ini adalah masalah yang paling sulit diatasi dalam praktek aplikasi kunci publik. Ini adalah semacam neraka bagi kriptografi kunci publik, dan banyak kompleksitas aplikasi termakan dalam memecahkan masalah ini. Kamu seharusnya menggunakan kunci publik jika setelah kamu yakin bahwa kunci publik tersebut bagus dan tidak pernah dipalsukan, dan kunci tersebut benar-benar kepunyaan orang yang memang berasosiasi dengan sender. Kamu dapat memastikan ini jika kamu mendapatkan sertifikat kunci publik dari pemiliknya secara langsung, atau jika terdapat signature dari seseorang yang kamu percaya. Dan juga, user ID harus mengandung nama lengkap dari pemilik kunci, tidak hanya nama pertamanya. Dalam keadaan apapun, jangan sekali-kali anda mempercayaikunci publik yang anda download dari bulletin board, kecuali jika kunci tersebut ditandai oleh orang yang anda percayai. Kunci publik yang belum sah dapat saja berasal dari kunci asli yang dipalsukan oleh orang lain, bahkan dapat juga merupakan hasil pemalsuan dari sistem administrator dari bulletin board tersebut.

Jika kamu diminta untuk memberi tanda pada kunci publik orang lain, maka pastikan bahwa orang tersebut benar-benar yang berwenang dengan melihat nama orang pada user ID dari kunci publik tersebut. Hal ini karena signature yang anda berikan pada kunci publik orang adalah sebuah janji oleh anda bahwa kunci publik ini adalah benar-benar milik orang yang bersangkutan. Orang lain yang percaya terhadap anda akan menerima kunci publik orang tersebut karena mengandung signature dari anda. Jadi dapat dikatakan bahwa jangan sekali-kali menandai kunci publik seseorang sebelum anda benar-benar memiliki pengetahuan bahwa kunci tersebut adalah benar-benar milik seseorang tersebut.



Lebih baik lagi jika anda memberi tanda, jika kamu mendapatkan langsung dari pemilik pesan. Dalam kasus penandaan kunci publik, anda harus lebih memastikan tentang kepemilikan kunci dibandingkan dengan kepentingan anda untuk menggunakan kunci tersebut dalam melakukan enkripsis pesan. Untuk lebih meyakinkan dari validitas kunci tersebut cukup terpercaya, sertifikat dari introducer yang terpercaya sudah mencukupi. Tapi untuk menandai kunci sendiri, kamu harus memiliki pengetahuan yang pasti tentang siapa pemilik kunci yang sebenarnya.

Diharapkan anda melakukan panggilan ke pemilik kunci melalui telepon dan membacakan kuncinya, untuk memastikan kunci yang anda miliki adalah benar-benar kuncinya dan pastikan juga anda berbicara dengan orang yang benar di telepon tersebut. Suatu ide bagus untuk menyimpan kunci publik kita dengan koleksi dari certifying signatures yang diberikan dari bermacam variasi dari “introducers”, dengan harapan bahwa banyak orang akan percaya dengan paling tidak satu introducers yang meragukan validitas dari kunci publik anda. Anda dapat mengirimkan kunci dengan koleksi

dari certifying signatures dalam beberapa electronic bulletin boards. Jika kamu menandai kunci publik orang lain, maka kembalikan kepada dia dengan signature anda sehingga orang yang bersangkutan dapat menambahan signature anda ke koleksi dari signature untuk kunci publiknya. Dan pastikan bahwa tidak ada orang lain yang dapat memalsukan dengan menggunakan public keyring anda. Mengecek kunci publik yang baru harus berdasarkan integritas dari kunci publik yang terpercaya yang telah ada pada public keyring anda. Menjaga kontrol fisik dari public keyring anda lebih baik dilakukan pada personal computer anda dibandingkan dengan melalui remote time-sharing system, seperti juga yang anda lakukan terhadap kunci privat. Hal ini lebih ditujukan untuk melindungi kunci dari pencurian, bukanlah untuk menghindari penyingkapan. Jagalah kopi backup dari public keyring dan private key anda dalam media yang write-protected. Jika kunci publik anda digunakan untuk authority final untuk secara langsung ataupun tidak langsung certify semua kunci-kunci yang lain yang ada pada keyring anda, maka kunci tersebut merupakan kunci yang paling penting untuk diproteksi dari pencurian.

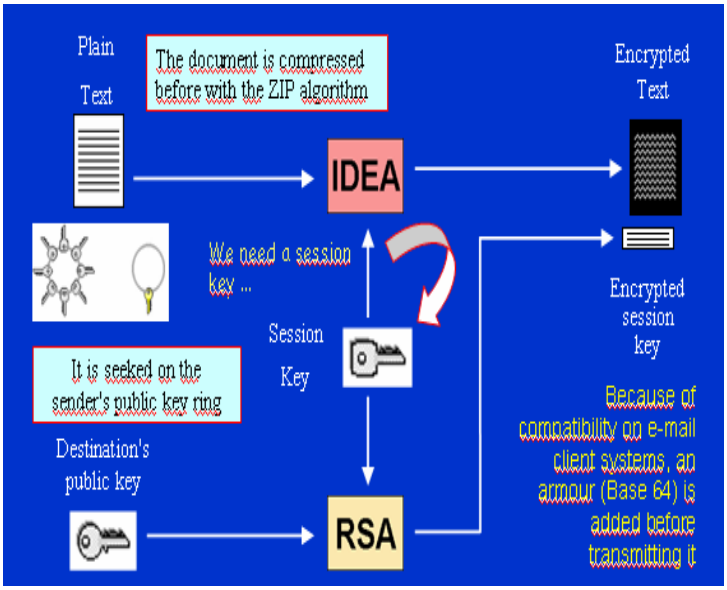
Anda mungkin menginginkan untuk membuat backupnya dalam write-protected floppy disk. PGP secara umum mengasumsikan jika kamu akan menjaga kemanan fisik untuk sistem anda dan juga keyring anda, demikian jug dengan kopi dari PGP itu sendiri. Sesuatu yang complicated untuk memproteksi seluruh public keyring kita dari pencurian adalah dengan memberi tanda seluruh ring dengan private key kita. Anda dapat melakukan ini dengan membuat detached signature pada public keyring.

2.2.4 Cara PGP Mengetahui Kunci yang Valid

PGP menjaga track record dari kunci mana dari public keyring anda yang certified dengan menggunakan signature dari introducer yang anda percayai. Yang anda lakukan adalah memberitahukan kepada PGP orang mana yang kamu percaya sebagai introducers dan certify kunci mereka dengan kunci sendiri yang anda percayai. PGP dapat mengambilnya dari situ, secara otomatis melakukan validasi kunci-kunci lain yang diberikan oleh introducers. Dan tentu saja anda dapat secara langsung memberi tanda pada kunci yang lain dengan mandiri. Ada dua kriteria terpisah yang PGP gunakan untuk memberi keputusan pada kunci publik :

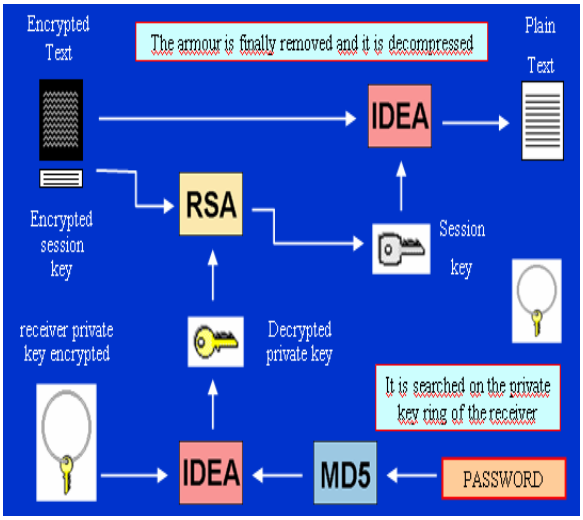
1. Apakah kunci tersebut telah disetifikasi oleh siganture yang terpercaya
2. Apakah kunci tersebut ada pada seseorang yang anda percayai untuk memberi signature pada kunci yang lain

PGP dapat menghitung jawaban untuk pertanyaan pertama. Untuk menjawab pertanyaan yang kedua, anda harus memberitahukan pada PGP secara eksplisit. Ketika kamu memberi jawaban pada pertanyaan kedua, PGP kemudian akan menkalkulasi jawaban untuk pertanyaan pertama untuk kinci lain yang diberi tanda oleh introducer. Kunci yang telah disertifikasi oleh introducer yang terpercaya akan dinyatakan valid oleh PGP. Kunci yang berhubungan dengan introducer yang terpercaya harus disertifikasi oleh anda ataupun oleh introducer yang terpercaya. PGP juga memperbolehkan kemungkinan anda memiliki bermacam-macam kepercayaan terhadap orang untuk bertindak sebagai introducer. Kepercayaan anda terhadap pemilik kunci untuk bertindak sebagai introducer tidak merefleksikan estimasi dari personal integrity. Tetapi ini seharusnya merefleksikan seberapa kompeten orang tersebut mengerti tentang key management dan menggunakan pemilihan keputusan yang tepat dalam pemilihan kunci.



Kamu dapat memutuskan orang sebagai yang terpercaya, terpercaya sebagian, ataupun terpercaya total untuk melakukan sertifikasi kunci publik yang lain. Informasi kepercayaan ini disimpan dalam keyring bersamaan dengan kuncinya, tapi ketika kamu ketika anda

melakukan perintah untuk melakukan kopi terhadap keyring, PGP tidak melakukan kopi terhadap trust information bersamaan dengan kuncinya, karena opini anda terhadap kepercayaan sangatlah penting.

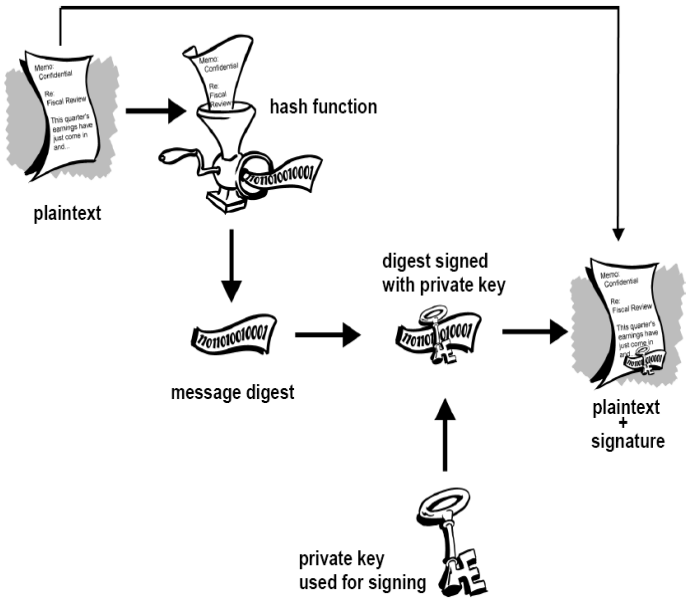


2.2.5 Menjaga *Private Key* dari Penyingkapan

Jagalah private kry anda dan passphare dengan sangat hati-hati. Jika private key anda pernah diomongkan, maka anda lebih baik langsung menggantinya sebelum kunci privat anda digunakan oleh orang lain untuk membuat signature dari nama anda. Sebagai contoh, seseorang dapat menggunakan kunci itu untuk memberi tanda kunci publik palsu yang tersertifikat, dimana dapat menyebabkan masalah bagi banyak orang, terutama jika signature anda sangat dipercaya. Dan tentu saja jika kunci private anda diketahui, maka dapat membuka semua pesan terenkripsi yang dikirimkan ke anda. Untuk melakukan proteksi terhadap kunci privat, anda dapat memulainya dengan selalu mengawasi kontrol fisik dari kunci tersebut.

Menyimpannya dalam personal computer di rumah bukan merupakan masalah, atau menyimpannya dalam notebook sehingga dapat anda bawa selalu. Jika kamu harus menggunakan komputer kantor dimana anda tidak memiliki kontrol fisik terhadap komputer tersebut, maka simpan public dan private keyring dalam write-protected removable floppy disk, dan jangan tinggalkan sembarangan ketika kamu meninggalkan kantor computer. Dan akan merupakan ide yang bagus untuk memungkinkan private key anda untuk diletakkan pada remote time-sharing computer, seperti remote dial-in UNIX system.

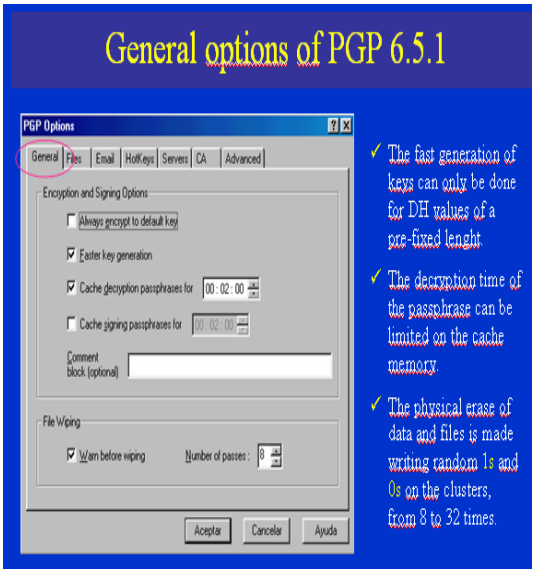
Seseorang dapat mengecek pada modem line anda dan mengambil passphare anda dan lalu mengamati private key yang benar melalui remote system. Anda sebaiknya hanya menggunakan kunci privat anda dalam mesin yang berada di bawah kontrol fisik anda. Jangan menyimpan passphare anda dimanapun dalam komputer yang mempunyai file private key anda. Menyimpan keduanya baik private key dan passphare pada komputer yang sama sangatlah berbahaya seperti halnya menyimpan PIN dalam dompet yang sama dengan kartu ATM anda. Anda pasti tidak menginginkan seseorang mengotak-atik disk anda yang di dalamnya mengandung passphare dan private key anda.



Dan akan lebih aman jika anda hanya mengingat passphare anda dan jangan menyimpannya dimana-mana kecuali hanya di pikiran anda. Jika anda merasa harus menulis passphare anda, maka proteksilah dengan baik, bahkan jika bisa proteklah dengan lebih baik daripada proteksi file private key. Dan jagalah backup dari private key anda. Pendekatan decentralized noninstitutional yang didukung oleh PGP untuk manajemen kunci publik mempunyai keuntungan, tapi sayangnya anda tidak dapat meletakkan dalam centralized list tunggal yang menyimpan daftar kunci.

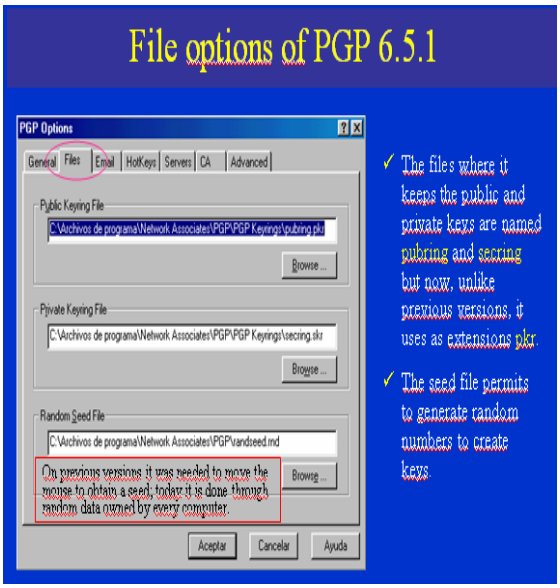
3. *PGP dalam Windows*

3.1 PGP versi 6.5.1

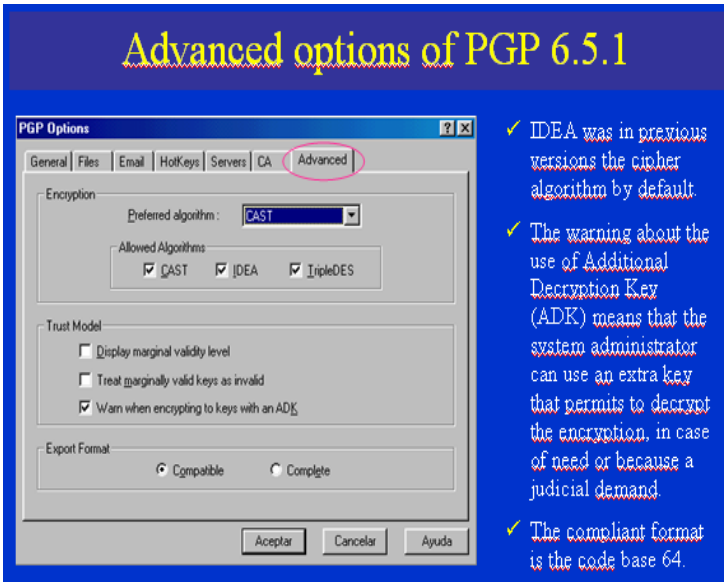


Gambar di atas menunjukkan pilihan umum dari PGP versi 6.5.1.

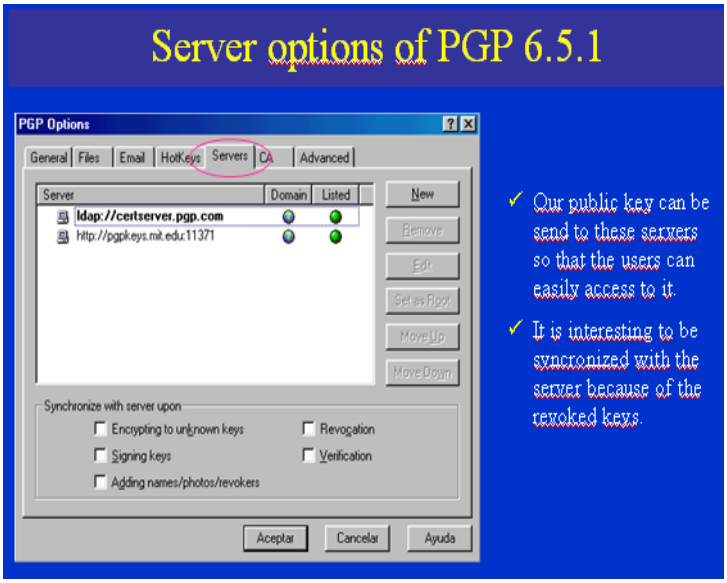
Sedangkan pilihan file dari PGP versi 6.5.1 adalah sebagai berikut :



Untuk pemilihan metode enkripsi yang digunakan, diatur dalam option advance seperti berikut ini

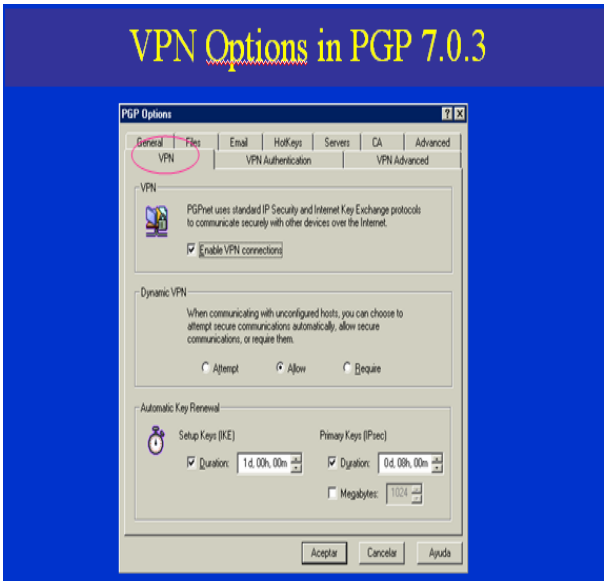


Gambar di bawah ini menunjukkan langkah-langkah pemilihan server yang akan digunakan :



3.2 PGP versi 7.0.3

Terdapat sedikit perbedaan antara PGP 7.0.3 dengan versi 6.5.1 yaitu bahwa pada PGP 7.0.3 disediakan mode VPN (Virtual Private Network)



Selain penambahan fitur VPN, PGP versi 7.3.0 memiliki kelebihan yang lain yaitu :

1. Disediakan fasilitas berupa dekripsi otomatis.
2. Memperkenalkan dua algoritma baru : AES dan Twofish
3. Pembentukan kunci dilakukan otomatis oleh DH/DSS dengan panjang 1024 bit. Jika kita ingin membuat RSA keys, kita harus masuk ke expert option sebagai mandatory

DAFTAR PUSTAKA

[1] Aguirre, Jorge Ramió. (2005). Electronic Book About Computer Security and Crypthography. *http://www.criptored.upm.es/guiateoria/gt_m001a.htm*. Tanggal akses:6 Oktober 2006 pukul 21:00.

[2] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[3] *NAI*. (1999). Network Associates, Inc. *http://www.nai.com*. Tanggal akses: 6 Oktober 2006 pukul 20:00.

[4] Kak, Avinash. (2006). Security for Internet Application. Purdue University.