BAB II
Dasar-Dasar Keamanan Sistem Informasi

Pendahuluan

Sebelum melangkah lebih jauh kepada hal yang praktis dalam pengamanan sistem informasi, ada baiknya kita pelajari dasar-dasar (*principles*) dan teori-teori yang digunakan untuk pengamanan sistem informasi. Kriptografi, enkripsi, dan dekrtipsi (baik dengan menggunakan private-key maupun dengan menggunakan public-key) akan dibahas di dalam bab ini.

Pengamanan informasi (dengan menggunakan enkripsi) memiliki dampak yang luar biasa dimana hidup atau mati seseorang sangat bergantung kepadanya. Mungkin contoh nyata tentang hal ini adalah terbongkarnya pengamanan informasi dari Mary, Queen of Scots[1], sehingga akhirnya dia dihukum pancung. Terbongkarnya enkripsi yang menggunakan Enigma juga dianggap memperpendek perang dunia kedua. Tanpa kemampuan membongkar Enkripsi mungkin perang dunia kedua akan berlangsung lebih lama dan korban perang akan semakin banyak.

Terminologi

- **Kriptografi (*cryptography) merupakan ilmu dan seni untuk menjaga pesan*** agar aman. (*Cryptography is the art and science of keeping messages secure. [40]) "Crypto" berarti "secret" (rahasia) dan "graphy" berarti" writing" (tulisan) [3]. Para pelaku atau praktisi kriptografi disebut **cryptographers. Sebuah algoritma kriptografik (cryptographic algorithm),*** disebut **cipher, merupakan persamaan matematik yang digunakan untuk** proses enkripsi dan dekripsi. Biasanya kedua persamaan matematik (untuk enkripsi dan dekripsi) tersebut memiliki hubungan matematis yang cukup erat..

Proses yang dilakukan untuk mengamankan sebuah pesan (yang disebut *plaintext*) menjadi pesan yang tersembunyi (disebut *ciphertext*) adalah **enkripsi** (*encryption*). *Ciphertext* adalah pesan yang sudah tidak dapat dibaca dengan mudah. Menurut ISO 7498-2, terminologi yang lebih tepat digunakan adalah "*encipher*".

Proses sebaliknya, untuk mengubah *ciphertext* menjadi *plaintext*, disebut **dekripsi** (*decryption*). Menurut ISO 7498-2, terminologi yang lebih tepat untuk proses ini adalah "*decipher*".

Cryptanalysis adalah seni dan ilmu untuk memecahkan *ciphertext* tanpa bantuan kunci. *Cryptanalyst* adalah pelaku atau praktisi yang menjalankan *cryptanalysis*. *Cryptology* merupakan gabungan dari *cryptography* dan *cryptanalysis*.

Kriptografi dan Sistem Informasi

- Keamanan untuk Sistem Informasi
- Informasi ditujukan untuk segolongan tertentu
- SI dengan kriptografi
- 4 Aspek Fundamental dari Sistem Kriptografi
  - Kerahasiaan (Confidentiality)
  - Integritas Data (Data Integrity)
  - Otentifikasi (Authentication)
  - Ketiadaan Penyangkalan (Non-Repudiation)

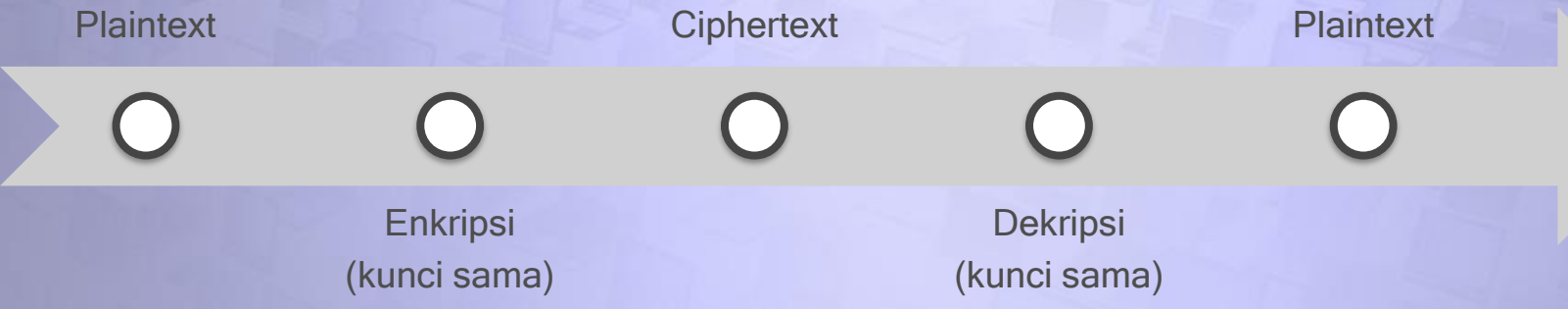
Mekanisme Kriptografi

- Mekanisme sistem kriptografi bekerja dengan cara menyandikan pesan menjadi kode rahasia yang dimengerti oleh pelaku sistem informasi saja.
- Istilah umum yang digunakan pada kriptografi :
  - Plaintext
  - Chiphertext
  - Cipher
  - Enkripsi
  - Dekripsi
  - Kriptosistem

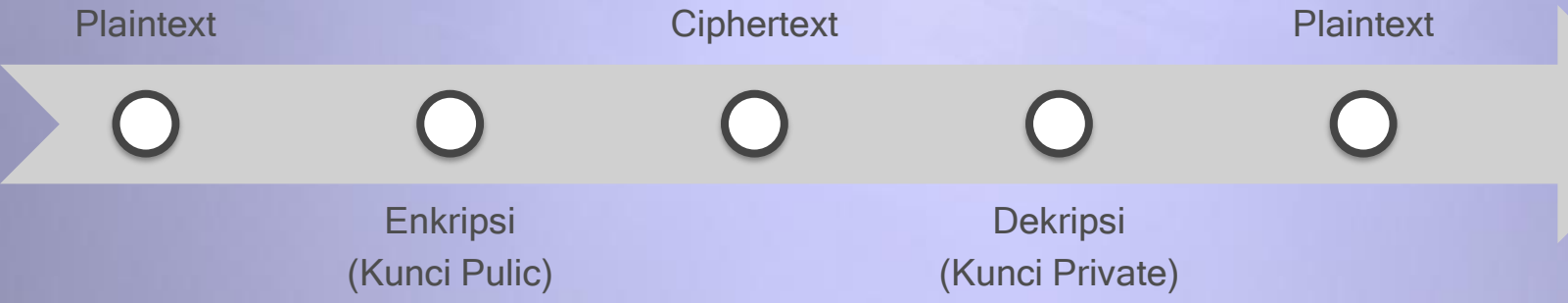
Enkripsi

- Enkripsi Dibentuk berdasar suatu algoritma yang akan mengacak suatu informasi menjadi bentuk yang tidak dapat dibaca / di lihat
- Dekripsi adalah proses dengan algoritma yang sama untuk mengembalikan pesan kebentuk aslinya
- Algoritma yang digunakan harus terdiri dari susunan prosedur yang direncanakan secara hati-hati dan efektif.
- Muncul algoritma kriptografi Simetris, Asimetris dan Single Way

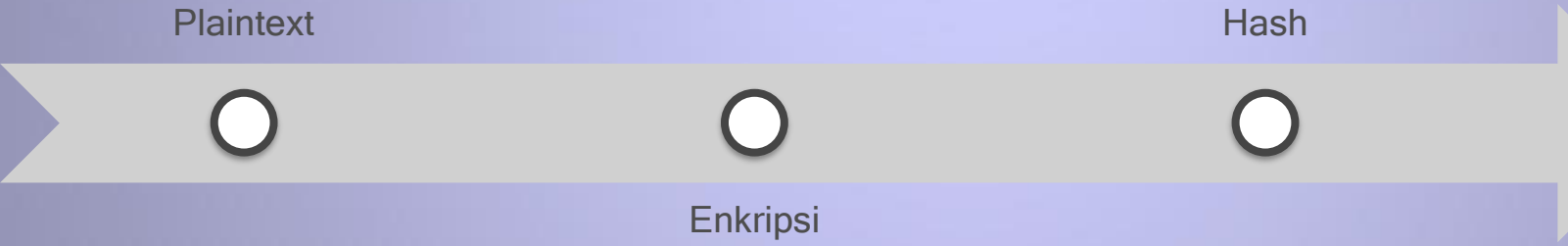
Enkripsi



Kunci Simetris



Kunci ASimetris



One Way Function

Fungsi Matematika Enkripsi

Secara matematis, proses atau fungsi enkripsi (E) dapat dituliskan sebagai:

$$E(M) = C \quad (1)$$

dimana: M adalah *plaintext* (*message*) dan C adalah *ciphertext*.

Proses atau fungsi dekripsi (D) dapat dituliskan sebagai:

$$D(C) = M \quad (2)$$

Elemen Ekripsi

Algoritma dari Enkripsi dan Dekripsi. Algoritma dari enkripsi adalah fungsi-fungsi yang digunakan untuk melakukan fungsi enkripsi dan dekripsi. Algoritma yang digunakan menentukan kekuatan dari enkripsi, dan ini biasanya dibuktikan dengan basis matematika.

Berdasarkan cara memproses teks (*plaintext), cipher dapat dikategorikan* menjadi dua jenis: *block cipher and stream cipher.* ***Block cipher*** *bekerja* dengan memproses data secara blok, dimana beberapa karakter / data digabungkan menjadi satu blok. Setiap proses satu blok menghasilkan keluaran satu blok juga. Sementara itu ***stream cipher*** bekerja memproses masukan (karakter atau data) secara terus menerus dan menghasilkan data pada saat yang bersamaan.

Elemen Ekripsi

Kunci yang digunakan dan panjangnya kunci. Kekuatan dari penyandian bergantung kepada kunci yang digunakan. Beberapa algoritma enkripsi memiliki kelemahan pada kunci yang digunakan. Untuk itu, kunci yang lemah tersebut tidak boleh digunakan. Selain itu, panjangnya kunci, yang biasanya dalam ukuran *bit*, juga menentukan kekuatan dari enkripsi. Kunci yang lebih panjang biasanya lebih aman dari kunci yang pendek. Jadi enkripsi dengan menggunakan kunci 128-bit lebih sukar dipecahkan dengan algoritma enkripsi yang sama tetapi dengan kunci 56-bit. Semakin panjang sebuah kunci, semakin besar keyspace yang harus dijalani untuk mencari kunci dengan cara *brute force attack* atau coba-coba karena keyspace yang harus dilihat merupakan pangkat dari bilangan 2. Jadi kunci 128-bit memiliki keyspace 2^{128}, sedangkan kunci 56-bit memiliki keyspace 2^{56}. Artinya semakin lama kunci baru bisa ketahuan.

Elemen Ekripsi

Kembali ke masalah algoritma, keamanan sebuah algoritma yang digunakan dalam enkripsi atau dekripsi bergantung kepada beberapa aspek. Salah satu aspek yang cukup penting adalah sifat algoritma yang digunakan. Apabila kekuatan dari sebuah algoritma sangat tergantung kepada pengetahuan (tahu atau tidaknya) orang terhadap algoritma yang digunakan, maka algoritma tersebut disebut "restricted algorithm". Apabila algoritma tersebut bocor atau ketahuan oleh orang banyak, maka pesan-pesan dapat terbaca. Tentunya hal ini masih bergantung kepada adanya kriptografer yang baik. Jika tidak ada yang tahu, maka sistem tersebut dapat dianggap aman (meskipun semu).

Meskipun kurang aman, metoda pengamanan dengan *restricted algorithm* ini cukup banyak digunakan karena mudah implementasinya dan tidak perlu diuji secara mendalam. Contoh penggunaan metoda ini adalah enkripsi yang menggantikan huruf yang digunakan untuk mengirim pesan dengan huruf lain. Ini disebut dengan "*substitution cipher*".

Elemen Ekripsi

Substitution Cipher dengan Caesar Cipher

Salah satu contoh dari "*substitution cipher*" adalah Caesar Cipher yang digunakan oleh Julius Caesar. Pada prinsipnya, setiap huruf digantikan dengan huruf yang berada tiga (3) posisi dalam urutan alfabet. Sebagai contoh huruf "a" digantikan dengan huruf "D" dan seterusnya. Transformasi yang digunakan adalah:

```
plain : a b c d e f g h i j k l m n o p q r s t u v w x y z
cipher: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

Latihan 1. Buat ciphertext dari kalimat di bawah ini.

PESAN SANGAT RAHASIA

Latihan 2. Cari plaintext dari kalimat ini

PHHW PH DIWHU WKH WRJD SDUWB

Elemen Ekripsi

ROT13

Substitution cipher yang masih umum digunakan di sistem UNIX adalah ROT13. Pada sistem ini sebuah huruf digantikan dengan huruf yang letaknya 13 posisi darinya. Sebagai contoh, huruf "A" digantikan dengan huruf "N", huruf "B" digantikan dengan huruf "O", dan seterusnya. Secara matematis, hal ini dapat dituliskan sebagai:

$$C = ROT13(M) \quad (3)$$

Untuk mengembalikan kembali ke bentuk semulanya dilakukan proses enkripsi ROT13 dua kali [37].

$$M = ROT13(ROT13(M)) \quad (4)$$

Elemen Ekripsi

Caesar cipher dan ROT13 disebut juga **"*monoalphabetic ciphers*"** *karena* setiap huruf digantikan dengan sebuah huruf. Huruf yang sama akan memikili pengganti yang sama. Misalnya huruf "a" digantikan dengan huruf "e", maka setiap huruf "a" akan digantikan dengan huruf "e". Mono alphabetic cipher ini agak mudah dipecahkan dengan menganalisa ciphertext apabila beberapa informasi lain (seperti bahasa yang digunakan) dapat diketahui. Salah satu cara penyerangan (*attack) yang dapat dilakukan* adalah dengan menganalisa statistik dari frekuensi huruf yang muncul. Cara ini disebut ***frequency analysis*** *[39]. Stallings dalam bukunya [40]* menunjukkan statistik kemunculan huruf untuk tulisan dalam bahasa Inggris, dimana huruf "e" yang paling banyak muncul. Cara yang sama dapat dilakukan untuk mencari distribusi penggunaan huruf dalam teks berbahasa Indonesia.

Frequency analysis bermanfaat jika teks yang tersedia cukup panjang. Teks yang pendek, dengan jumlah huruf yang lebih sedikit, biasanya memiliki deviasi dari data-data statistik munculnya huruf. Selain itu ada beberapa kasus dimana sengaja dibuat teks yang "merusak" struktur frekuensi tersebut. Sebagai contoh, pengarang Perancis yang bernama Georges Perec di tahun 1969 menulis "*La Disparition*" (sebuah novel dengan 200 halaman) tanpa kata yang menggunakan huruf "e". Karya ini kemudian diterjemahkan oleh ke dalam bahasa Inggris oleh seorang pengarang Inggris yang bernama Gilbert Adair dengan tetap tanpa menggunakan huruf "e". Judul terjemahannya adalah "*A Void*". Cerita ini diulas dalam buku [39].

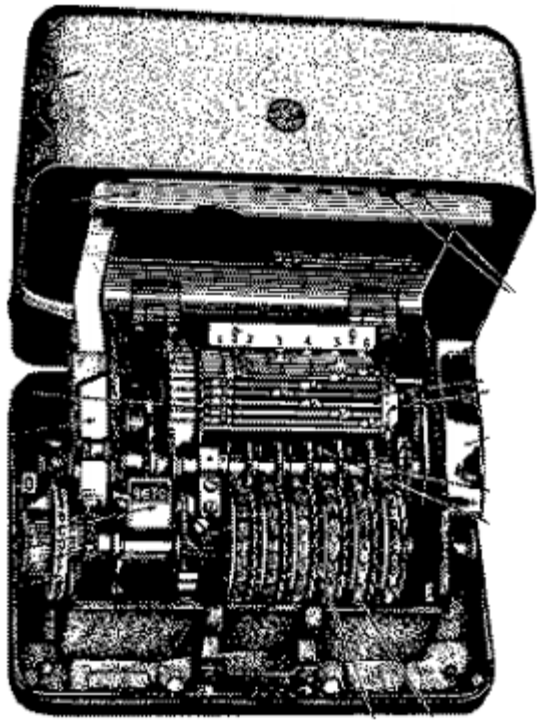
Elemen Ekripsi

Multiple-letter encryption

Untuk meningkatkan keamanan, enkripsi dapat dilakukan dengan mengelompokkan beberapa huruf menjadi sebuah kesatuan (unit) yang kemudian dienkripsi. Ini disebut *multiple-letter encryption*. Salah satu contoh multiple-letter encryption adalah "*Playfair*".

Enigma Rotor Machine

Enigma rotor machine merupakan sebuah alat enkripsi dan dekripsi mekanik yang digunakan dalam perang dunia ke dua. Dia terdiri atas beberapa rotor dan kabel yang silang menyilang menyebabkan substitusi alfabet yang selalu berubah sehingga Enigma mengimplementasikan polyalphabetic chiper. Buku "Code Book" [39] banyak membahas tentang Enigma ini.



Enigma Rotor Machine

Elemen Ekripsi

- **Aplikasi dari Enkripsi**

  Contoh penggunaan enkripsi adalah program Pretty Good Privacy (PGP) [15], dan secure shell (SSH). Program PGP digunakan untuk mengenkripsi dan menambahkan digital siganture dalam e-mail yang dikirim. Program SSH digunakan untuk mengenkripsi sesion telnet ke sebuah host. Hal ini akan dibahas lebih lanjut pada bagian lain.

Public-key Cryptography VS Symetric Cryptography

Perbedaan prinsip dan penggunaan *public-key cryptography* dan *symmetric cryptography* membutuhkan diskusi tersendiri. Pada *symmetric cryptography*, satu kunci yang sama digunakan untuk melakukan enkripsi dan dekripsi. Pada sistem *public-key cryptography*, enkripsi dan dekripsi menggunakan kunci yang berbeda.

Sejak dikembangkannya *public-key cryptography*, selalu timbul pertanyaan mana yang lebih baik. Para pakar kriptografi mengatakan bahwa keduanya tidak dapat dibandingkan karena mereka memecahkan masalah dalam domain yang berbeda. *Symmetric cryptography* merupakan hal yang terbaik untuk mengenkripsi data. Kecepatannya dan keamanan akan *choosen-ciphertext attack* merupakan kelebihannya. Sementara itu *public-key cryptography* dapat melakukan hal-hal lain lebih baik daripada *symmetric cryptography*, misalnya dalam hal key management. (Diskusi lebih jauh dapat dilihat di referensi [37].)

Elemen Ekripsi

Penggunaan Kunci

Salah satu cara untuk menambah tingkat keamanan sebuah algoritma enkripsi dan dekripsi adalah dengan menggunakan sebuah kunci (*key*) yang biasanya disebut *K*. Kunci *K* ini dapat memiliki rentang (*range*) yang cukup lebar. Rentang dari kemungkinan angka (harga) dari kunci *K* ini disebut *keyspace*. Kunci K ini digunakan dalam proses enkripsi dan dekripsi sehingga persamaan matematisnya menjadi:

$$E_K(M) = C$$

$$D_K(M) = M$$

Keamanan sistem yang digunakan kemudian tidak bergantung kepada pengetahuan algoritma yang digunakan, melainkan bergantung kepada kunci yang digunakan. Artinya, algoritma dapat diketahui oleh umum atau dipublikasikan. Usaha untuk memecahkan keamanan sistem menjadi usaha untuk memecahkan atau mencari kunci yang digunakan.

Usaha mencari kunci sangat bergantung kepada keyspace dari kunci *K*. Apabila keyspace ini cukup kecil, maka cara *brute force* atau mencoba semua kunci dapat dilakukan. Akan tetapi apabila keyspace dari kunci yang digunakan cukup besar, maka usaha untuk mencoba semua kombinasi kunci menjadi tidak realistis. Keyspace dari *DES*, misalnya, memiliki 56-bit. Untuk mencoba semua kombinasi yang ada diperlukan 2^{56} kombinasi.

Latihan 5. Jika sebuah komputer dapat mencoba 1000 kombinasi dalam 1 detik, berapa waktu yang dibutuhkan untuk mencoba semua kombinasi DES yang menggunakan 56 bit?

Elemen Ekripsi

Data Encryption Standard (DES)

DES, atau juga dikenal sebagai *Data Encryption Algorithm* (DEA) oleh ANSI dan DEA-1 oleh ISO, merupakan algoritma kriptografi simetris yang paling umum digunakan saat ini. Sejarahnya DES dimulai dari permintaan pemerintah Amerika Serikat untuk memasukkan proposal enskripsi. DES memiliki sejarah dari Lucifer[1], enkripsi yang dikembangan di IBM kala itu. Horst Feistel merupakan salah satu periset yang mula-mula mengembangkan DES ketika bekerja di IBM Watson Laboratory di Yorktown Heights, New York. DES baru secara resmi digunakan oleh pemerintah Amerika Serikat (diadopsi oleh National Bureau of Standards) di tahun 1977. Ia dikenal sebagai Federal Information Processing Standard 46 (FIPS PUB46).

Aplikasi yang menggunakan DES antara lain:

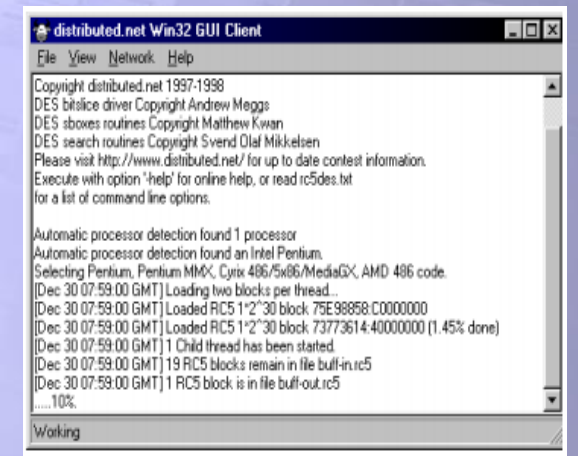
- enkripsi dari password di sistem UNIX
- berbagai aplikasi di bidang perbankan

Elemen Ekripsi

Memecahkan DES

DES merupakan block chiper yang beroperasi dengan menggunakan blok berukuran 64-bit dan kunci berukuran 56-bit. Brute force attack dengan mencoba segala kombinasi membutuhkan 2^{56} kombinasi atau sekitar $7x\ 10^{17}$ atau 70 juta milyar kombinasi.

DES dengan penggunaan yang biasa (*cookbook mode*) dengan panjang kunci 56 bit saat ini sudah dapat dianggap tidak aman karena sudah berhasil dipecahkan dengan metoda coba-coba (*brute force attack*).



GAMBAR 2.2. Contoh peraqaan client distributed.net untuk Windows 95

Bahan bacaan DES

Banyak sudah buku, artikel yang memuat informasi tentang DES. Bagi anda yang berminat untuk mempelajari DES lebih lanjut, silahkan mengunakan referensi [11, 13, 24, 27, 37 - Chapter 12].

Untuk DES cracker dari EFF, silahkan kunjungi web sitenya di http://www.eff.org/descracker.html

Ada berbagai group yang mencoba memecahkan DES dengan berbagai cara. Salah satu group yang bernama ***distributed.net*** menggunakan teknologi Internet untuk memecahkan problem ini menjadi sub-problem yang kecil (dalam ukuran blok). Pengguna dapat menjalankan sebuah program yang khusus dikembangkan oleh tim ini untuk mengambil beberapa blok, via Internet, kemudian memecahkannya di komputer pribadinya. Program yang disediakan meliputi berbagai operating system seperti Windows, DOS, berbagai variasi Unix, Macintosh. Blok yang sudah diproses dikembalikan ke *distributed.net* via Internet. Dengan cara ini puluhan ribu orang, termasuk penulis, membantu memecahkan DES. Mekanisme ini dapat memecahkan DES dalam waktu 30 hari.

Elemen Ekripsi

Hash function - integrity checking

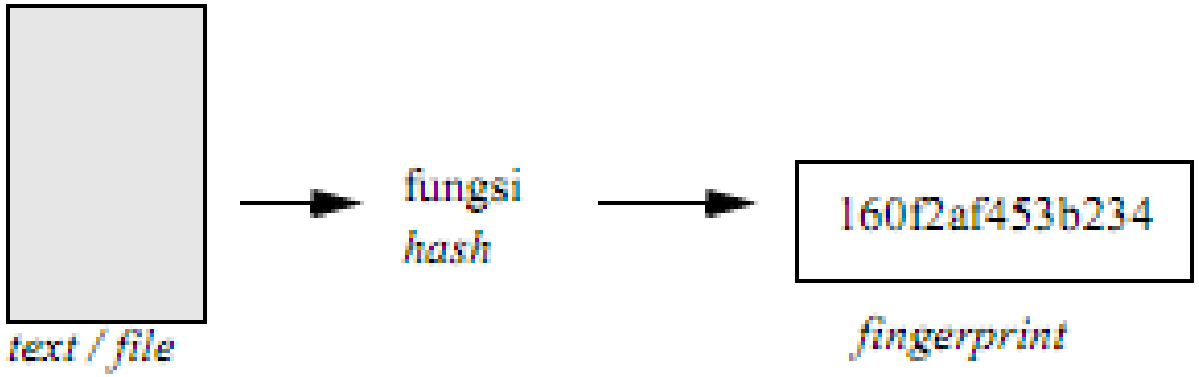
Salah satu cara untuk menguji integritas sebuah data adalah dengan memberikan "checksum" atau tanda bahwa data tersebut tidak berubah. Cara yang paling mudah dilakukan adalah dengan menjumlahkan karakter-karakter atau data-data yang ada sehingga apabila terjadi perubahan, hasil penjumlahan menjadi berbeda. Cara ini tentunya mudah dipecahkan dengan menggunakan kombinasi data yang berbeda akan tetapi menghasilkan hasil penjumlahan yang sama.

Pada sistem digital biasanya ada beberapa mekanisme pengujian integritas seperti antara lain:

- parity checking
- checksum
- hash function

Hash function merupakan fungsi yang bersifat satu arah dimana jika kita masukkan data, maka dia akan menghasilkan sebuah "checksum" atau "fingerprint" dari data tersebut.

- MD5
- SHA



Elemen Ekripsi

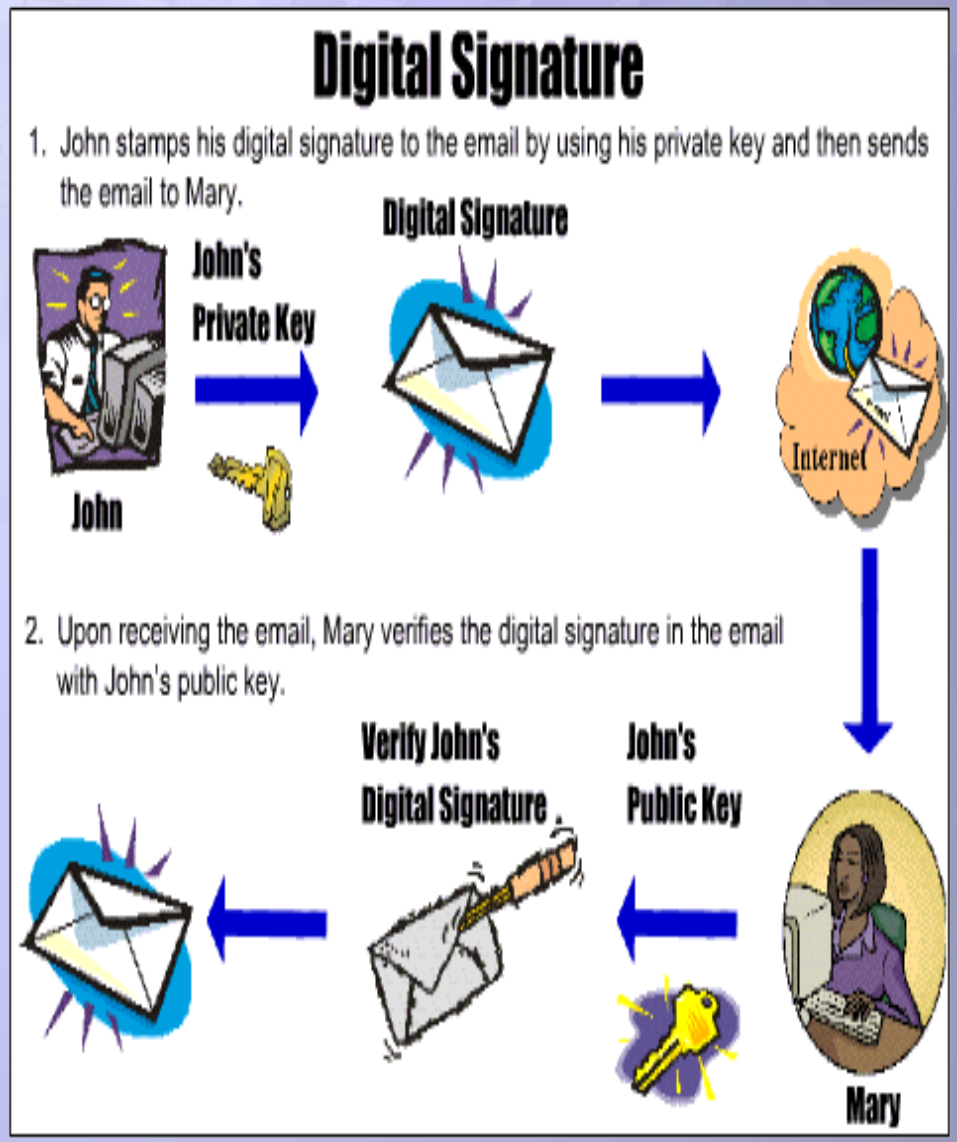
Latihan 6. Gunakan MD5 untuk menghasilkan fingerprint dari kalimat berikut: "Saya pesan 10 buah komputer." (tanpa tanda petik). Kemudian nbadingkan hasil MD5 dengan kalimat: "Saya pesan 11 buah komputer."

Contoh latihan di atas dapat dijalankan pada sistem UNIX yang memiliki program "md5" (atau program "md5sum"[1]) seperti di bawah ini.

```
unix% echo 'Saya pesan 10 buah komputer.' | md5
5F736F18556E3B8D90E50299C7345035
unix% echo 'Saya pesan 11 buah komputer.' | md5
9CB9AD1A369512C96C74236B959780D3
```

Contoh di atas menunjukkan bahwa perbedaan satu karakter saja sudah menghasilkan keluaran hash yang sangat berbeda. Hasil yang serupa dapat dilakukan dengan menggunakan SHA (Secure Hash Algorithm) atau algoritma dan program lainnya.

Fungsi hash ini biasanya digabungkan dengan enkripsi untuk menjaga integritas. Sebagai contoh, dalam pengiriman email yang tidak rahasia (dapat dibaca orang) akan tetapi ingin dijaga integritasnya, isi (*body*) dari email dapat dilewatkan ke fungsi hash sehingga menghasilkan fingerprint dari isi email tersebut. Keluaran dari hash ini dapat disertakan dalam email. Ketika email diterima, penerima juga menjalankan fungsi hash terhadap isi email dan kemudian membandingkannya dengan hash yang dikirim. Jika email diubah di tengah jalan, maka kedua hash tersebut berbeda. Untuk lebih meningkatkan keamanan, hasil dari hash juga dapat dienkripsi sehingga hanya penerima saja yang dapat membuka hasil dari hash tersebut. Atau dapat juga hasil hash dienkripsi dengan kunci privat pengirim sehingga oleh penerima dapat dibuka dengan kunci publik pengirim dan diyakinkan bahwa integritas dari isi terjamin serta pengirim betul-betul berasal dari pemilik kunci publik tersebut. Inilah yang sering disebut digital signature dalam email.



Merci bien
ありがとう
Matur Nuwun
Hatur Nuhun
Obrigado
Dank
Thanks
Matur se Kelangkong
Syukron
Kheili Mamnun
ευχαριστίες
Danke
Grazias
谢谢
Terima Kasih


Irawan_afrianto@yahoo.com


Irawan.afrianto


@Irawan_afrianto


+628170223513

TUGAS I (Individu)

- Buat Makalah mengenai Peran / Implementasi Tanda Tangan Digital
- Referensi Min 10 Paper Terkait Tanda tangan Digital
- Memuat : Judul, Pendahuluan, Teori Pendukung, Gagasan Penerapan Tanda tangan digital
- Tema tanda tangan digital untuk Indonesia
- Dikumpulkan Saat UTS