BAB II

TINJAUAN PUSTAKA

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. Crypto berarti *secret* (rahasia) dan graphia berarti *writing* (tulisan). Menurut terminologinya, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. (Dony Ariyus, 2008, 13)

Pada pengertian modern, kriptografi adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas (Rifki Sadikin, 2012, 9).

Pada dasarnya komponen kriptografi terdiri dari beberapa komponen, seperti (Dony Ariyus, 2008, 10) :

1. *Enkripsi*: merupakan hal yang sangat penting dalam kriptografi, merupakan cara pengamanan data yang dikirimkan sehingga terjaga kerahasiaannya. Pesan asli disebut *plaintext* (teks-biasa), yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *cipher* atau kode. Seperti halnya dengan tidak mengerti sebuah kata maka dapat dicari artinya di dalam kamus atau daftar istilah. Beda halnya dengan enkripsi, untuk mengubah teks-biasa ke bentuk teks-kode yang menggunakan algoritma yang dapat mengkodekan data yang diingini.
2. *Dekripsi*: merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya. Walaupun dekripsi kebalikan dari enkripsi tetapi algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunakan untuk enkripsi.

3. *Kunci*: adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, yaitu kunci rahasia (*private key*) dan kunci umum (*public key*).

4. *Ciphertext*: merupakan suatu pesan yang telah melalui proses enkripsi. Pesan yang ada pada teks-kode ini tidak dapat dibaca karena berupa karakter-karakter yang tidak mempunyai makna (arti).

5. *Plaintext*: sering disebut dengan *cleartext*. Teks-asli atau teks-biasa ini merupakan pesan yang ditulis atau diketik yang memiliki makna. Teks-asli inilah yang diproses menggunakan algoritma kriptografi untuk ciphertext.

6. *Pesan*: dapat berupa data atau informasi yang dikirim (melalui kurir, saluran komunikasi data, dsb) atau yang disimpan di dalam media perekaman.

7. *Cryptanalysis*: Kriptanalisis bisa diartikan sebagai analisis kode atau suatu ilmu untuk mendapatkan teks-asli tanpa harus mengetahui kunci yang sah secara wajar. Jika suatu teks-kode berhasil diubah menjadi teks-asli tanpa menggunakan kunci yang sah, proses tersebut dinamakan *breaking code*.

2.1.1 Kriptografi Klasik

Kriptografi klasik umumnya merupakan teknik penyandian dengan kunci simetrik dan menyembunyikan pesan yang memiliki arti ke sebuah pesan yang nampaknya tidak memiliki arti dengan metode subsitusi dan transposisi. (Rifki Sadikin, 2012, 15)

Kriptografi klasik memiliki beberapa ciri, yaitu (Doni Ariyus, 2008, 49):

1. Berbasis karakter
2. Menggunakan pena dan kertas saja, belum ada komputer
3. Termasuk ke dalam kriptografi kunci simetri

2 Tipe operasi yang dipakai dalam enkripsi dan dekripsi (Rifki Sadikin, 2012, 17):

1. Subsitusi, elemen pada pesan (karakter, *byte*, atau *bit*) ditukar/ disubsitusi dengan elemen lain dari ruang pesan. Misalnya subsitusi sederhana A ditukar B, B ditukar D, dan C ditukar Z, pesan "BACA" menjadi "DBZB";
2. Transposisi, elemen pada pesan berpindah posisi misalnya posisi 1 menjadi posisi 4 dan posisi 2 menjadi posisi 3, posisi 3 menjadi posisi 1 dan posisi 4 menjadi posisi 2, pesan "KAMI" menjadi "MAIK".

2.1.2 Kriptografi Modern

Berbeda dengan penyandian klasik yang umumnya berorientasi pada karakter, penyandian modern berorientasi bit sebab penyandian modern menggunakan media komputer untuk mengolah pesan. Pesan pada sandi modern tidak selalu berupa rangkaian karakter bisa saja berupa rangkaian bit seperti berkas video atau berkas gambar.

Terdapat 2 jenis operasi sandi modern yaitu:

1. Sandi *stream*, yang beroperasi pada data *stream* sehingga operasi penyandian dilakukan per satu bit atau per satu *byte* pada satu waktu.
2. Sandi blok, biasa mengolah teks asli sebagai satu kesatuan (dengan ukuran tertentu) dan menghasilkan teks sandi dengan ukuran yang sama.

Baik sandi blok atau sandi *stream* mempunyai wilayah aplikasinya sendiri-sendiri. (Rifki Sadikin, 2012, 95)

Pada kriptografi modern terdapat berbagai macam algoritma yang dimaksudkan untuk mengamankan informasi yang dikirim melalui jaringan komputer, Algoritma modern terdiri dari dua bagian (Doni Ariyus, 2008, 108):

1. Algoritma Simetris, adalah algoritma yang menggunakan kunci yang sama untuk enkripsi dan dekripsinya. Aplikasi dari algoritma simetris digunakan oleh beberapa algoritma seperti : *Data Encryption Standard* (DES), *Advanced Encryption Standard* (AES), *International Data Encryption Algoritma* (IDEA), A5, RC4.
2. Algoritma Asimetris, adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan yang satu lagi untuk dekripsi. Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA.

Algoritma kriptografi yang dibahas pada penulisan ini adalah algoritma RSA, karena algoritma tersebutlah yang digunakan untuk menambah lapisan keamanan gambar atau citra digital.

2.2 Algoritma Rivest Shamir Adleman

Dari sekian banyak algoritma kriptografi kunci-publik yang pernah dibuat, algoritma yang paling popular adalah algoritma Rivest Shamir Adleman(RSA). Algoritma ini melakukan pemfaktoran bilangan yang sangat besar. Oleh karena alasan tersebut RSA dianggap aman. Untuk membangkitkan dua kunci, dipilih dua bilangan prima acak yang besar. Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (*Massachussets Institute of Technology*) pada tahun 1976, yaitu Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman. RSA mengekspresikan teks-asli yang dienkripsi menjadi blok-blok yang mana setiap blok memiliki nilai bilangan biner yang diberi symbol “n”, blok teks-asli “M” dan blok teks-kode “C”. Untuk melakukan enkripsi pesan “M”, pesan dibagi ke dalam blok-blok numeric yang lebih kecil daripada “n” (data biner dengan pangkat terbesar). Jika bilangan prima yang panjangnya 200 digit, dapat

ditambahkan beberapa bit 0 di kiri bilangan untuk menjaga agar pesan tetap kurang dari nilai "n". (Dony Ariyus, 2008, 148)

Besaran yang digunakan pada algoritma RSA (Dony Ariyus, 2008, 149) :

1. p dan q bilangan prima (rahasia)
2. r = p * q (rahasia)
3. Φ(r) = (p – 1)(q – 1) (rahasia)
4. PK (kunci enkripsi) (tidak rahasia)
5. SK (kunci dekripsi) (rahasia)
6. X (teks-asli) (rahasia)
7. Y (teks-kode) (tidak rahasia)

2.2.1 Notasi Matematika Untuk RSA

Untuk memahami algoritma RSA, seseorang harus memahami beberapa notasi matematika dasar, teori dan formula. Hal tersebut dibutuhkan untuk mendukung semua kalkulasi yang dilakukan dalam algoritma RSA. (*http://ezine.echo.or.id/ezine12/echo12-05.txt*)

a. Modulo (didenotasikan dengan 'x mod m' atau 'x % m' dalam beberapa bahasa komputer)

- x % m = x mod m = pembagian x dengan m dan mengambil sisanya.
- Contoh: 25 mod 5 = 0 karena 5 habis membagi 25
- 25 mod 4 = 1 karena 25 / (4 * 6) menyisakan 1
- x mod m = x jika dan hanya jika x < m

b. GCD(A,B)

Greatest Common Divisor atau sering disingkat GCD adalah operasi yang sering digunakan dalam manipulasi bilangan bulat (bilangan yang tidak

memiliki angka dibelakang koma, bukan tergolong bilangan riil) dan banyak digunakan dalam banyak operasi terapan misalnya dalam ilmu kriptografi dan *hash table*. GCD atau juga dikenal sebagai FPB (Faktor Persekutuan terBesar), mencari nilai factor pembagi bersama yang paling besar dari dua nilai masukkan.

Misalnya : GCD(80, 12)

- Faktor pembagi dari 80 adalah **1**, **2**, **4**, 5, 8, 10, 16, 20, 40 dan 80 itu sendiri
- Faktor pembagi dari 12 adalah **1**, **2**, 3, **4**, 6 dan 12 itu sendiri
- Faktor pembagi bersama untuk nilai 80 dan 12 adalah 1, 2 dan 4
- Dari faktor pembagi bersama tersebut yang terbesarnya adalah 4
- Jadi GCD(80, 12) = 4

Ada cara lain untuk mencari nilai GCD, selain dengan cara mencari masing-masing faktor pembagi dan kemudian menentukan factor pembagi bersamanya dan mengambil nilai yang terbesar, yaitu dengan menggunakan algoritma *Euclidean*. Algoritma *Euclidean* dapat dijelaskan dengan algoritma abstrak berikut ini.

Misalkan: GCD (m, n)

1. Selama n tidak sama dengan 0 lakukan langkah 2 dan langkah 3, namun jika sama dengan nol maka langsung kerjakan langkah 4
2. Ambil sisa dari pembagian m dengan nilai n, dan simpan di r
3. Ganti nilai m lama dengan n lama dan ganti nilai n yang lama dengan r (sisa bagi), kemudian kerjakan lagi langkah 2
4. Pada langkah ini nilai m adalah GCD-nya asalkan nilai n sudah bernilai 0.

2.2.2 Rumusan Algoritma RSA

Terdapat 3 algoritma pada sistem kriptografi RSA, yaitu algoritma pembangkitan kunci, algoritma enkripsi, dan algoritma dekripsi. (Rifki Sadikin, 2012, 250)

Algoritma RSA didasarkan pada teorema *Euler* yang menyatakan bahwa nilai $a^{\Phi(n)} \equiv 1 \pmod{n}$ yang dalam hal ini:

1. a harus relatif prima terhadap n atau gcd(a, n) = 1
2. $\Phi(n) = n(1-1/p1)(1-1/p2) \ldots (1-1/pn)$, yang dalam hal ini p1, p2, …, pn adalah factor prima dari n. $\Phi(n)$ adalah fungsi yang menentukan berapa banyak bilangan 1, 2, 3, …, n yang relatif prima terhadap n.

Berdasarkan persamaan $(X^e)^d \equiv X \pmod{r}$ maka enkripsi dan dekripsi dirumuskan sebagai berikut:

1. $E_e(X) = Y \equiv X^e \pmod{n}$, rumus untuk enkripsi
2. $D_d(Y) = X \equiv Y^d \pmod{n}$, rumus untuk dekripsi

Sebelum melakukan enkripsi dan dekripsi terlebih dahulu dilakukan pembangkitan kunci dengan langkah-langkah sebagai berikut:

1. p dan q merupakan bilangan prima. Hitung n = p x q
2. Hitung nilai $\Phi(n) = (p-1) \times (q-1)$
3. Menentukan nilai e yang merupakan kunci umum untuk enkripsi. Nilai e harus relatif prima terhadap $\Phi(n)$ atau $GCD(e, \Phi(n)) = 1$
4. Nilai d merupakan kunci rahasia untuk dekripsi. Nilai d didapatkan dengan persamaan $e \times d = 1 + k \times \Phi(n)$ atau $d = (1 + k \times \Phi(n))/e$

Contoh:

Misalkan Childva mengirim pesan “HELLO WORLD” kepada Chitra dengan nilai numerik pesan adalah 07 04 11 11 14 26 22 14 17 11 03. Dengan p = 7, q = 11, n = p x q = 77, dan Φ(n) = (p-1) x (q-1) = 60. Kemudian Childva memilih kunci umum e = 17, dan syarat memenuhi karena GCD(17, 60) = 1. Lalu pasangan kunci rahasia d = 53 yang didapat dari k = 15 dan d = (1 + k x Φ(n))/e = (1 + 15 x 60)/17. Childva melakukan enkripsi dengan kunci umum untuk menghasilkan *ciphertext*. Kemudian Chitra melakukan dekripsi dengan kunci rahasia untuk menghasilkan *plaintext*. Sebagai ilustrasi dapat dilihat pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.1 Ilustrasi Enkripsi dari Algoritma RSA (Rifki Sadikin, 2012, 250)

ENKRIPSI			
Teks Asli (X)	Desimal (X)	$Y=X^e \pmod n$	Teks kode (Y)
H	7	$7^{17} \bmod 77$	28
E	4	$4^{17} \bmod 77$	16
L	11	$11^{17} \bmod 77$	14
L	11	$11^{17} \bmod 77$	14
O	14	$14^{17} \bmod 77$	42
	26	$26^{17} \bmod 77$	38
W	22	$22^{17} \bmod 77$	22
O	14	$14^{17} \bmod 77$	42
R	17	$17^{17} \bmod 77$	19
L	11	$11^{17} \bmod 77$	44
D	13	$13^{17} \bmod 77$	75

Tabel 2.2 Ilustrasi Dekripsi dari Algoritma RSA(Rifki Sadikin, 2012, 250)

Dekripsi			
Teks kode (Y)	$X=Y^d \pmod n$	Desimal (X)	Teks Asli (X)
28	$28^{53} \bmod 77$	7	H
16	$16^{53} \bmod 77$	4	E
14	$14^{53} \bmod 77$	11	L
14	$14^{53} \bmod 77$	11	L
42	$42^{53} \bmod 77$	14	O
38	$38^{53} \bmod 77$	26	
22	$22^{53} \bmod 77$	22	W

Tabel 2.2 Ilustrasi Dekripsi dari Algoritma RSA(Rifki Sadikin, 2012, 250)(Lanjutan)

Dekripsi			
Teks kode (Y)	$X=Y^d \pmod n$	Desimal (X)	Teks Asli (X)
42	$42^{53} \bmod 77$	14	O
19	$19^{53} \bmod 77$	17	R
44	$44^{53} \bmod 77$	11	L
75	$75^{53} \bmod 77$	13	D

2.2.3 Letak Keamanan Pada RSA

Keamanan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan non-prima menjadi faktor primanya, yang dalam hal ini r = p x q. Sekali r berhasil difaktorkan menjadi p dan q maka Φ(r) = (p-1)(q-1) dapat dihitung. Selanjutnya, karena kunci enkripsi PK diumumkan (tidak rahasia) maka kunci dekripsi SK dapat dihitung dari persamaan PK . SK ≡ 1 (mod Φ(r)).

Penemu algoritma RSA menyarankan agar panjang nilai p dan q lebih dari 100 digit. Dengan demikian hasil kali r = p x q akan lebih dari 200 digit. (Doni Ariyus, 2008, 162).