BAB II

LANDASAN TEORI

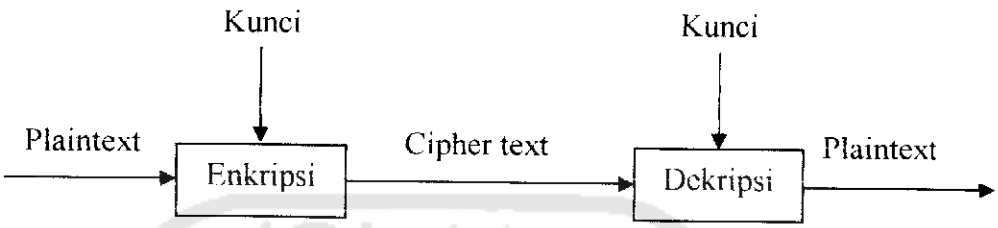
2.1 Kriptografi

Kriptografi (*cryptography*) merupakan ilmu dan seni penyimpanan pesan, data, atau informasi secara aman. Kriptografi (*Cryptography*) berasal dari bahasa Yunani yaitu dari kata *Crypto* dan *Graphia* yang berarti penulisan rahasia. Kriptografi adalah suatu ilmu yang mempelajari penulisan secara rahasia. Kriptografi merupakan bagian dari suatu cabang ilmu matematika yang disebut *Cryptology*. Kriptografi bertujuan menjaga kerahasiaan informasi yang terkandung dalam data sehingga informasi tersebut tidak dapat diketahui oleh pihak yang tidak sah.

Dalam menjaga kerahasiaan data, kriptografi mentransformasikan data jelas (*plaintext*) ke dalam bentuk data sandi (*ciphertext*) yang tidak dapat dikenali. Ciphertext inilah yang kemudian dikirimkan oleh pengirim (*sender*) kepada penerima (*receiver*). Setelah sampai di penerima, ciphertext tersebut ditranformasikan kembali ke dalam bentuk plaintext agar dapat dikenali kembali.

Proses tranformasi dari plaintext menjadi ciphertext disebut proses *Encipherment* atau enkripsi (*encryption*), sedangkan proses mentransformasikan kembali ciphertext menjadi plaintext disebut proses dekripsi (*decryption*). Untuk mengenkripsi dan mendekripsi data, kriptografi menggunakan suatu algoritma (cipher) dan kunci (key). Cipher adalah fungsi matematika yang digunakan untuk

mengenkripsi dan mendekripsi. Sedangkan kunci merupakan sederetan bit yang diperlukan untuk mengenkripsi dan mendekripsi data. Lihat gambar 2.1.



Gambar 2.1. Proses Enkripsi/Dekripsi Sederhana

Cryptography adalah suatu ilmu ataupun seni mengamankan pesan, dan dilakukan oleh *cryptographer*. Sedang, *cryptanalysis* adalah suatu ilmu dan seni membuka (*breaking*) ciphertext dan orang yang melakukannya disebut *cryptanalyst*.

Cryptographic system atau *cryptosystem* adalah suatu fasilitas untuk mengkonversikan plaintext ke ciphertext dan sebaliknya. Dalam sistem ini, seperangkat parameter yang menentukan transformasi pencipheran tertentu disebut suatu set kunci. Proses enkripsi dan dekripsi diatur oleh satu atau beberapa kunci kriptografi. Secara umum, kunci-kunci yang digunakan untuk proses pengenkripsian dan pendekripsian tidak perlu identik, tergantung pada sistem yang digunakan.

Secara umum operasi enkripsi dan dekripsi dapat diterangkan secara matematis sebagai berikut :

EK (M) = C (Proses Enkripsi)

DK (C) = M (Proses Dekripsi)

Pada saat proses enkripsi kita menyandikan pesan M dengan suatu kunci K lalu dihasilkan pesan C. Sedangkan pada proses dekripsi, pesan C tersebut diuraikan dengan menggunakan kunci K sehingga dihasilkan pesan M yang sama seperti pesan sebelumnya.

Dengan demikian keamanan suatu pesan tergantung pada kunci ataupun kunci-kunci yang digunakan, dan tidak tergantung pada algoritma yang digunakan. Sehingga algoritma-algoritma yang digunakan tersebut dapat dipublikasikan dan dianalisis, serta produk-produk yang menggunakan algoritma tersebut dapat diproduksi massal. Tidaklah menjadi masalah apabila seseorang mengetahui algoritma yang kita gunakan. Selama ia tidak mengetahui kunci yang dipakai, ia tetap tidak dapat membaca pesan.

2.2 Algoritma Kriptografi

Berdasarkan kunci yang dipakai, algoritma kriptografi dapat dibedakan atas dua golongan, yaitu :

1. Algoritma Simetris
2. Algoritma Asimetris

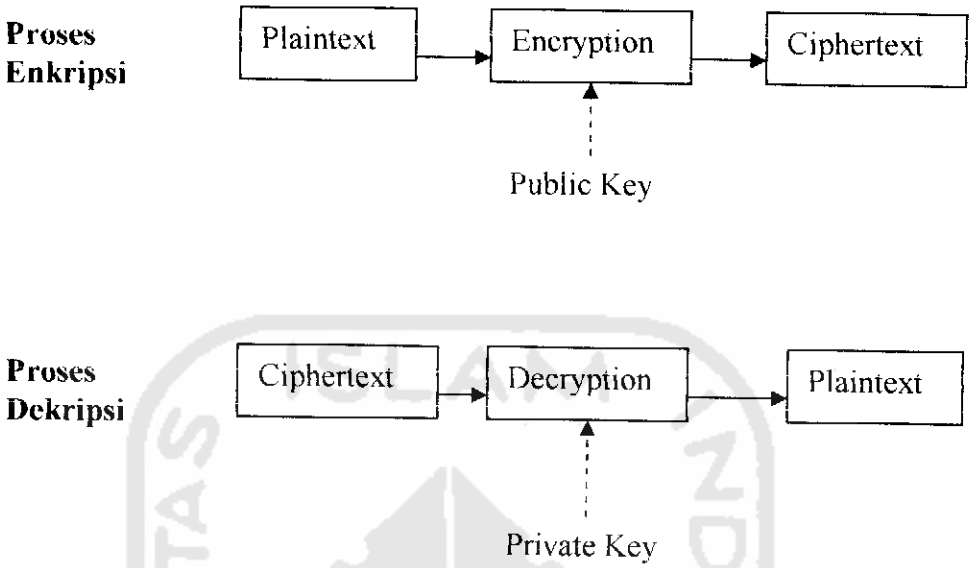
2.2.1 Algoritma Simetris

Algoritma kriptografi simeteris atau disebut juga algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi.

Algoritma kriptografi simetcris dibagi menjadi 2 kategori yaitu algoritma aliran (*Stream Ciphers*) dan algoritma blok (*Block Ciphers*). Pada algoritma aliran, proses penyandiannya berorientasi pada satu bit atau satu byte data. Sedang pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan bit atau byte data (per blok). Contoh algoritma kunci simetris yang terkenal adalah DES (*Data Encryption Standard*).

2.2.2 Algoritma Asimetris

Algoritma kriptografi asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Algoritma ini disebut juga algoritma kunci umum (*public key algorithm*) karena kunci untuk enkripsi dibuat umum (*public key*) atau dapat diketahui oleh setiap orang, tapi kunci untuk dekripsi hanya diketahui oleh orang yang berwenang mengetahui data yang disandikan atau sering disebut kunci pribadi (*private key*). Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA, PGP dan ECC. Untuk proses eknkripsi dan dekripsi dengan kunci publik dapat dilihat pada gambar 2.2.



Gambar 2.2. Proses Enkripsi/Dekripsi kriptografi kunci publik

2.3 Algoritma RSA

Sistem kriptografi RSA adalah salah satu sistem kriptografi kunci publik yang ditemukan oleh *Rivest, Shamir dan Adleman.* Sejak skema sistem ini ditemukan, sistem ini menguasai sebagai satu-satunya sistem yang diterima dan diterapkan secara luas sebagai sistem kriptografi kunci publik. Sistem ini termasuk sistem enkripsi blok, karena data asli dan data sandi adalah bilangan integer antara 0 sampai (n -1), untuk semua nilai n positif [STA95].

RSA merupakan sebuah terobosan baru dalam sistem enkripsi data, kita sebelumnya mengenal enkripsi data dimana cara untuk meng-enkripsi dan cara men-dekripsi sebuah data dilakukan dengan kunci yang sama sebagai contoh enkripsi data sederhana semisal kita punya data ABC dan dienkripsi menjadi BCD (tiap huruf diganti menjadi huruf yang ada pada urutan berikutnya) maka saat

teman kita akan mendekrip data BCD kembali menjadi ABC, cara ini lah yang disebut juga enkripsi symetric dimana algoritma dalam mendekripsi dan meng-enkripsi mempunyai kesamaan algoritma (tinggal membalik proses/simetris).

RSA hadir dengan cara baru yaitu enkripsi yang asimetris (algoritma untuk mengenkrip dan men-dekrip merupakan dua hal yang berbeda namun mempunyai hasil yang sama). teknik RSA ini disebut juga sistem enkripsi dengan public key - private key. berikut adalah konsep pemikiran dari RSA :

Dianggap semua orang sudah mempunyai 2 buah kunci (kode untuk enkripsi dan dekripsi seterusnya akan disebutkan sebagai sebuah "kunci") , kedua kunci ini yang satu dinamakan Private key dan yang satu lagi dinamakan Public key, private key sesuai namanya maka harus disimpan / hanya diketahui oleh si pemilik kemudian public key diberikan ke orang lain atau ditaruh pada sebuah sumber yang bebas seperti Internet.

2.3.1 Landasan Matematis untuk Algoritma RSA

Pada subbab ini akan dibahas landasan matematis dari algoritma RSA, termasuk teori-teori dan algoritma yang melandasi perhitungan.

2.3.1.1 Bilangan Prima

Bilangan prima adalah bilangan bulat > 1 yang hanya habis dibagi 1 san bilangan itu sendiri. Manusia telah mengenal bilangan prima sejak 6500 SM. Tulang Ishango yang ditemukan pada tahun 1960 (sekarang disimpan di musee d'histoire naturelle di brussel) membuktikan hal tersebut. Tulang Ishango

memiliki 3 baris takik. Salah satu kolomnya memiliki 11, 13, 17 dan 19 takik, yang merupakan bilangan prima antara 10 hingga 20.

Meskipun sedikit sekali manfaat yang diketahui, namun di awal masehi orang tetap mencari dan membuktikan bahwa suatu bilangan merupakan bilangan prima. Cara yang paling efesien untuk mencari bilangan prima kecil (misalkan kurang dari 10^7) adalah dengan menggunakan metode *Serve of Eratosthenes* (240 SM) sebagi berikut: daftarlah semua bilangan bulat antara 2 hingga n. Hapuslah semua bilangan kelipatan bilangan prima yang lebih kecil atau sama dengan $\sqrt{n}$, maka bilangan yang masih tersisa adalah bilangan prima.

Sebagai contoh, untuk mencari semua bilangan prima ≤ 30, pertama-tama didaftarkan semua bilangan bulat antar 2 hingga 30

2 3 4 5 6 7 8 9 10 11 12 13 14

15 16 17 18 19 20 21 22 23 24 25 26 27

28 29 30

Bilangan pertama (=2) adalah bilangan prima. Hapus semua bilangan kelipatan 2. (*bilangan mod 2 0*), maka didapat

2 3 5 7 9 11 13 15 17 19 21 23 25

27 29

Bilangan prima setelah 2 dalam daftar tersebut adalah 3, yang merupakan bilangan prima kedua. Hapus semua bilangan kelipatan 3 dari daftar. Didapat:

2 3 5 7 11 13 17 19 23 25 29

Bilangan prima setelah 3 dalam daftar tersebut adalah 53. Hapus semua bilangan kelipatan 5 dari daftar. Didapat:

2 3 5 7 11 13 17 19 23 29

Bilangan yang tidak terhapus berikutnya adalah 7 yang kuadratnya = 49 > 30. maka bilangan yang tersisa dalam daftar merupakan himpunan semua bilangan prima < 30

2.3.1.2 Aritmatika Modulus

Aritmetika modulus (*modulus arithmetic*) yang juga dikenal dengan aritmetika jam (*clock arithmetic*), merupakan dasar bagi banyak algoritma kriptografi, termasuk RSA.

Pada tahun 1801 Carl F. Gauss memperkenalkan konsep kongruensi, yaitu Dua buah bilangan bulat a dan b disebut kongruen modulo n, atau diekspresikan sebagai

$$a \equiv b(\mathrm{mod}\, n) \tag{2.1}$$

jika dan hanya jika a b k n untuk k berupa bilangan bulat.

Untuk lebih memahaminya, berikut diberikan sebuah contoh,

$$27 \equiv 3(\mathrm{mod}\, 12) \tag{2.2}$$

karena 27 - 3 = 24 dapat dinyatakan sebagai 2 × 12 (k = 2). Dalam hal ini juga dapat dinyatakan bahwa,

$$27 \bmod 12 = 3, \tag{2.3}$$

yaitu sisa pembagian 27 terhadap 12 adalah 3.

2.3.1.3 Faktor Pembagi Bersama Terbesar

Faktor pembagi bersama terbesar (*greatest common divisor* – *gcd*) dari dua buah bilangan *a* dan *b* adalah bilangan terbesar yang dapat membagi kedua bilangan tersebut. Sebagai contoh,

gcd (10, 15) = 5

gcd (18, 10) = 2

gcd (10, 21) = 1

Greatest common divisor dari dua buah bilangan tidak lain adalah irisan dari himpunan faktor bilangan prima dari kedua bilangan tersebut. Contohnya,

gcd (10, 15) = 5

gcd ((2 × 5), (3 × 5)) = 5

Ketika kedua bilangan tidak memiliki faktor bersama, maka gcd nya menjadi 1. Dalam hal ini, kedua bilangan disebut prima relatif (*relatively prime*). Pada contoh di atas, bilangan 10 dan 21 adalah prima relatif. Karena bilangan prima tidak mempunyai faktor lain selain dirinya sendiri, maka bilangan prima adalah relatif prima terhadap bilangan lain, kecuali kelipatannya. Salah satu cara untuk menghitung gcd dari dua buah bilangan adalah dengan menggunakan algoritma Euclid.

2.3.1.4 Invers Modulo

Seperti halnya aritmetika lainnya, dalam aritmetika modulus dikenal pula invers, yang dinamakan invers modulo. Bila dapat dinyatakan

$$A \cdot b \equiv 1 \pmod{n} \qquad (2.4)$$

atau

$$(a \cdot b) \bmod n = 1 \tag{2.5}$$

maka invers dari a, modulo n adalah b.

Invers modulo dari suatu bilangan seringkali tidak unik. Sebagai contoh, bila dinyatakan

$$4 \cdot x \equiv 1 \pmod 7, \tag{2.6}$$

maka x adalah invers dari 4, modulo 7. Ada beberapa buah bilangan x yang memenuhi pernyataan tersebut, yaitu x = 2, 9, 16, dan seterusnya karena

$$(4 \cdot 2) \bmod 7 = 8 \bmod 7 = 1$$

$$(4 \cdot 9) \bmod 7 = 36 \bmod 7 = 1$$

$$(4 \cdot 16) \bmod 7 = 64 \bmod 7 = 1$$

Beberapa bilangan bahkan tidak memiliki invers modulo. Secara umum dalam selang [0, n] tertentu, sebuah bilangan a memiliki invers yang unik b, modulo n, hanya jika a dan n prima relatif. Sebagai contoh, 5 dan 21 adalah prima relatif karena gcd (5, 21) = 1, maka invers modulo 21 dari 5 pada selang [0, 21] adalah unik, yaitu 17.

2.3.1.5 Teorema Fermat

Teorema Fermat dipublikasikan pada tahun 1640, namun baru dibuktikan hampir seratus tahun kemudian oleh teorema Euler. Teorema Fermat adalah sebagai berikut,

Jika p adalah bilangan prima dan a prima relatif terhadap p sehingga gcd (a, p) 1, maka

$$a^{p-1} \equiv 1 (\text{mod } p) \tag{2.7}$$

2.3.1.6 Teorema Euler

Pada tahun1736, Euler memperkenalkan sebuah fungsi yang dinamakan fungsi totient Euler (*Euler's totient function*) yang diberi notasi ϕ. Definisi dari fungsi ini adalah sebagai berikut,

Fungsi totient Euler dari suatu bilangan n ($n \geq 1$), yang diberi notasi $\phi(n)$, menyatakan jumlah bilangan bulat positif yang lebih kecil dari n dan prima relatif terhadap n.

Sebagai contoh, berikut ini diberikan nilai fungsi totient Euler untuk n = 1, 2, 3, 4, dan 5.

- $\phi(1) = 0$, tidak ada bilangan yang lebih kecil dari 1 dan prima relatif terhadap 1
- $\phi(2) = 1$, ada satu bilangan yang lebih kecil dari 2 dan prima relatif terhadap 2, yaitu 1
- $\phi(3) = 2$, bilangan yang lebih kecil dari 3 dan prima relatif terhadap 3 adalah 1 dan 2
- $\phi(4) = 2$, bilangan yang lebih kecil dari 4 dan prima relatif terhadap 4 adalah 1 dan 3
- $\phi(5) = 4$, bilangan yang lebih kecil dari 5 dan prima relatif terhadap 5 adalah 1,2,3, dan 4

Dapat dilihat bahwa nilai fungsi totient Euler untuk bilangan prima *p* adalah *(p 1)*, karena bilangan prima hanya memiliki dua faktor pembagi yaitu 1 dan dirinya sendiri. Untuk kasus perkalian bilangan prima bahkan fungsi totient Euler berlaku komposit, yaitu

(i) Jika n = p q dan p dan q adalah bilangan prima, maka (2.8)

(ii) $\phi(n) = \phi(p)\,\phi(q) = (p-1)(q-1)$ (2.9)

Pada tahun 1736 Euler telah mengajukan pembuktian terhadap teorema Fermat. Tetapi baru pada tahun 1760 Euler memberikan versi yang lebih umum dari teorema Fermat, yang kemudian dikenal sebagai bentuk umum Euler atas teorema Fermat (*Euler's generalization of Fermat's theorem*). Teorema ini menyatakan bahwa,

Jika $n \geq 2$ adalah bilangan bulat positif dan gcd (a, n) = 1, maka

$$a^{\phi(n)} = 1 \pmod{n}, \qquad (2.10)$$

di mana ϕ (n) menyatakan jumlah bilangan bulat positif yang prima relatif terhadap n

2.3.1.7 Bukti Matematis RSA

Setelah membahas semua latar belakang matematis dari algoritma RSA, berikut ini akan diberikan bukti matematis dari algoritma RSA. Bukti ini akan menunjukkan bahwa proses dekripsi benar-benar dapat menghasilkan plaintext semula dari ciphertext.

Karena persyaratan $e\,d \quad 1 \pmod{\phi}$, maka terdapat bilangan bulat positif k sehingga $e\,d \quad 1 \cdot k\,\phi$. Jika gcd $(m, p) = 1$ maka menurut teorema Fermat

$$m^{p-1} \equiv 1 \pmod{p} \tag{2.11}$$

Pemangkatan $k\ (q \quad 1)$ dan memperkalikan kedua sisi dengan m akan menghasilkan

$$m^{1 + k\,(p-1)\,(q-1)} = m \pmod{p} \tag{2.12}$$

Sementara, jika gcd $(m, p) = p$, maka kongruensi terakhir ini juga benar karena tiap sisinya kongruen terhadap 0 modulo p. Dengan demikian, untuk semua kasus berlaku

$$m^{1 \pmod{(p-1)\,(q-1)}} \equiv m \pmod{p} \tag{2.13}$$

$$m^{e\,d} \equiv m \pmod{p} \tag{2.14}$$

Dengan argumen yang sama dapat diperoleh pula

$$M^{e\,d} \equiv m \pmod{q} \tag{2.15}$$

Karena p dan q adalah dua bilangan prima yang berbeda dan $n \quad p\,q$, maka berlaku pula

$$M^{e\,d} \equiv m \pmod{n} \tag{2.16}$$

Dengan demikian dapat disimpulkan,

$$C^d = (m^{\,e})^{\,d} \equiv m \pmod{n} \tag{2.17}$$

2.3.2 Enkripsi dan Dekripsi Menggunakan RSA

Semisal A akan mengirimkan data atau pesan ke B maka A akan mengirimkan pesan ke B dengan terlebih dulu mengunci (mengekrip) dengan public key milik B sehingga hanya B yang dapat membuka (men-dekrip) pesan tersebut karena hanya B yang mempunyai pasangan kunci Public key B yaitu private key milik B schingga terjamin keamanannya.

2.3.3 Autentifikasi

Semisal A akan mengirimkan data atau pesan ke B dan A ingin agar B tahu bahwa A yang mengirim pesan tersebut maka A akan mengunci dengan private key milik A kemudian saat B menerima pesan A maka B dapat mengetahui isi pesan tersebut berasal dari A karena yang hanya dapat membuka pesan itu hanya kunci pasangan private key A yaitu Public key A yang sudah dimiliki B (ingat public key boleh dimiliki oleh semua orang).

Dari dua metode diatas dapat digabungkan menjadi metode yang tidak hanya secure tapi juga otentik.

Semisal A akan mengirimkan data ke B dan A ingin hanya B yang dapat membaca pesan itu serta B juga yakin bahwa yang mengirimakan pesan adalah A maka A mengunci pesan dengan private key milik A dan kemudian mengunci lagi dengan public key milik B kemudian data / pesan dikirimkan , B menerima pesan enkripsi dari A dan membuka dengan urutan Private key B dan kemudian Public Key A. sehingga data yang dikirimkan pasti secure karena hanya B yang dapat membaca pesan tersebut kemudian B pun tahu bahwa pasti A yang mengirimkan pesan tersebut cara ini disebut juga digital signature.

2.3.4 Perhitungan Matematis Algorithma RSA

Adapun cara perhitungan algoritma RSA adalah sebagai berikut:

1. Pilih dua nilai bilangan prima sembarang (p dan q)
2. Hitung n = pq
3. Hitung m = (p – 1) (q – 1) (teori euler)
4. Pilih nilai e

   Dengan kriteria bahwa e harus relatif prima terhadap m untuk perhitungan ini kita menggunakan perhitungan dengan euclid algorithm yaitu dengan greatest common divisor (gcd) bentuk perhitungan nya yaitu gcd (a,b) = gcd (b , a mod b). Dalam perhitungan pada contoh soal ini maka menggunakan gcd (e,m) bila hasil gcd (e,m) bukan nol maka nilai dari e tidak dapat digunakan karena tidak - relatively prime – kepada m
5. Hitung nilai d

   D = e -/ mod ϕ(n). Perhitungan dari d menggunakan extended euclid algorithm dari perhitungan maka nilai dari d = e -/ mod ϕ(n) equivalent dengan model de= 1 + nm dimana n adalah sebuah integer kemudian kita dapat menuliskan menjadi d = (1+nm)/e , dari sini kita akan melakukan perhitungan dengan smeua nilai n hingga nilai dari d ditemukan
6. Public key (e , n)
7. Private key (d , n)
8. Lakukan enkripsi dengan rumus? Hasil enkripsi = (dataasli dipangkat e) mod n

9. Lakukan dekripsi dengan rumus ? Hasil dekripsi = (hasilenkripsi dipangkat d) mod n

Perhatikan pula bahwa bila data dienkrip dengan public key maka harus dibuka/didekrip dengan private key dan hal yang sama juga di terapkan pada *digital signature* jika A mengirim B sebuah data dengan *digital signature* maka data akan dienkrip dengan private key A kemudian dengan public key B kemudian dikirimkan, saat B ingin membaca data asli maka B harus mendekrip dengan urutan private key B dan dilanjutkan dengan public key A.