KRIPTOGRAFI KUNCI PUBLIK

Revi Fajar Marta – NIM : 13503005

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail: if13005@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang kriptografi kunci publik dari berbagai segi, mulai dari konsep hingga penggunaannya. Kriptografi kunci publik berbeda dari kriptografi kunci simetri di mana hanya ada satu kunci rahasia untuk mengenkripsi/dekripsi pesan. Pada kriptografi kunci publik terdapat dua kunci, yaitu kunci publik dan kunci privat, yang salah satunya digunakan untuk mengenkripsi, sementara yang lainnya digunakan untuk mendekripsi pesan, dan mana yang digunakan untuk enkripsi/dekripsi tergantung aplikasinya.

Kriptografi kunci publik dapat digunakan dalam berbagai aplikasi, antara lain untuk kerahasiaan pesan, pembubuhan tanda tangan digital, serta pertukaran kunci simetri. Ketiganya dimaksudkan untuk menjaga keamanan dan keaslian data atau pesan, meskipun ada titik berat tertentu dalam tiap aplikasi. Penggunaan dalam kerahasiaan pesan dititikberatkan pada keamanan pesan dan data di dalamnya., tanda tangan digital dititikberatkan pada otentikasi data dan pemiliknya, sementara pertukaran kunci keduanya.

Penggunaan kriptografi kunci publik juga harus dilaksanakan dengan hati-hati, meskipun dapat digunakan dijalur komunikasi yang tidak aman, terutama karena penggunaannya pada suatu sistam biasanya melibatkan banyak pihak yang masing-masing harus terinformasikan mengenai pihak lainnya di dalm sistem tersebut. Ada 4 hal utama yang harus diperhatikan, yaitu hak atas pembangkitan kunci, pembuatan kunci baru untuk suatu pihak, penginformasian pihak lain atas pambuatan kunci baru, serta pemulihan dari kebocoran kunci. Semuanya dimaksudkan agar sistem berada dalam keadaan stabil dan aman.

Kata kunci: *Kriptografi kunci publik, public key cryptography*, *digital signature, key exchange algorithm*, enkripsi, dekripsi.

1. Pendahuluan

Kriptografi kunci publik, atau dikenal juga dengan sebutan kriptografi nirsimetri, adalah suatu bentuk kriptografi di mana seorang pengguna memiliki sepasang kunci kriptografi: kunci publik (public key) dan kunci privat (private key). Kunci privat disimpan secara rahasia, sementara kunci publik dapat disebarkan secara luas. Kedua kunci tersebut baerkaitan secara matematis, tetapi kunci privat tidak dapat diturunkan secara praktis dari kunci publik. Sebuah pesan yang dienkripsi dengan kunci publik hanya dapat didekripsi dengan kunci privat yang berpasangan dengannya.

Kebalikan dari kriptografi kunci publik, yaitu kriptografi kunci simetri menggunakan sebuah kunci rahasia untuk melakukan enkripsi dan dekripsi.

Dua cabang utama dari kriptografi kunci publik adalah:

1. Enkripsi kunci publik

   Sebuah pesan yang dienkripsi dengan kunci publik seseorang tidak dapat didekripsi oleh siapapun kecuali oleh pihak yang memiliki kunci privat pasangannya. Ini dimaksudkan untuk menjamin kerahasiaan data yang dienkripsi.

2. Tanda tangan digital

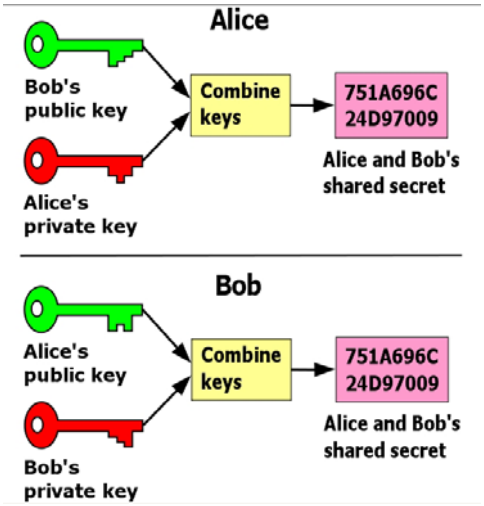
   Sebuah pesan yang ditandatangani menggunakan kunci privat seseorang dapat diverifikasi oleh siapapun yang memiliki akses terhadap kunci publik orang tersebut, sehingga membuktikan bahwa sang pemilik kunci telah menandatangani pesan tersebut dan bahwa pesan tersebut belum diubah oleh orang lain. Ini dimaksudkan untuk menjamin keaslian data yang dibubuhi tanda tangan digital.

Sebuah analogi dari enkripsi menggunakan kunci publik adalah sebuah kotak pos terkunci yang memiliki celah untuk memasukkan surat. Celah tersebut dapat diketahui dan dapat diakses oleh orang lain; lokasinya (alamat rumah) dapat dianalogikan sebagai kunci publik. Semua orang yang mengetahui alamat rumah orang tersebut dapat mendatangi kotak pos tersebut dan memasukkan surat ke dalamnya. Namun, hanya orang yang memiliki kunci kotak pos lah yang dapat membuka kotak surat dan membaca surat di dalamnya.

Sebuah analogi untuk tanda tangan digital adalah penyegelan sebuah amplop dengan lilin dan dibubuhi cap pribadi. Amplop tersebut dapat dibuka oleh siapapun. Keaslian surat tersebut ditandai dengan cap yang dibubuhkan di atas lilin.

Masalah utama dari kriptografi kunci publik adalah untuk membuktikan bahwa sebuah kunci publik otentik, dan belum diubah atau diganti oleh pihak ketiga. Pendekatan umum untuk mengatasi masalah ini adalah dengan menggunakan infrastruktur kunci publik (*public key infrastructure*; PKI), di mana satu atau lebih pihak ketiga, yang dikenal dengan *certification authority* (CA), menyertifikasi kepemilikan pasangan kunci. Pendekatan lainnya, yang digunakan oleh *Pretty Good Privacy* (PGP), adalah metode "*web of trust*" untuk menjamin keaslian pasangan kunci.

Metode kunci publik memiliki kerumitan komputasional melebihi algoritma kunci simetris. Pemakaian metode ini secara bijaksana memungkinkan banyak variasi aplikasi. Dalam prakteknya, kriptografi kunci publik dikombinasikan dengan metode kunci rahasia untuk alasan efisiensi (Gambar 1). Untuk enkripsi, pesan dapat dienkripsi dengan algoritma kunci rahasia menggunakan kunci yang dibangkitkan secara acak, dan kunci tersebut dienkripsi menggunakan kunci publik pengguna. Untuk tanda tangan digital, sebuah pesan di-*hash* (menggunakan fungsi hash, seperti MD5, SHA, dan sebagainya) dan nilai hash ditandatangani. Sebelum memverifikasi tanda tangan, penerima pesan menghitung terlebih dahulu nilai hash dari pesan yang diterimanya, lalu membandingkan hasilnya dengan nilai hash yang ditandatangani untuk mengecek keaslian pesan.



Gambar 1. Kunci rahasia dapat dicari dengan mengkalkulasikan kombinasi kunci privat milik pribadi dengan kunci publik milik orang lain. Kunci rahasia tersebut dapat digunakan sebagai kunci untuk cipher simetris.

Sejarah Kriptografi Kunci Publik

Sebenarnya penemu pertama kriptografi nirsimetri adalah James H. Ellis, Clifford Ccks, dan Malcolm Williamson di Inggris pada awal 1970. Mereka menemukan mekanisme pertukaran kunci, yang kemudian dikenal dengan nama algoritma pertukaran kunci Diffie-Hellman. Sayangnya algoritma mereka tersebut dirahasiakan dan tidak pernah dipublikasikan hingga tahun 1991.

Sistem kriptografi nirsimetri dipublikasikan pertama kali pada tahun 1976 oleh Whitfield Diffie dan Martin Hellman, dua orang ilmuwan dari Stanford University melalui makalah pertamanya di jurnal IEEE yang berjudul "New Directions in Cryptography". Makalah mereka membahas distribusi kunci rahasia pada saluran komunikasi publik (yang tidak aman) dengan metode pertukaran kunci Diffie-Hellman.

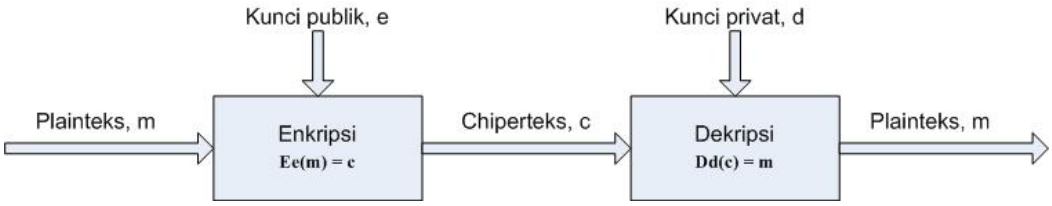
Pada tahun 1977, generalisasi dari ide Cocks ditemukan kembali oleh tiga orang ilmuwan dari MIT, yaitu Rivest, Shamir, dan Adleman. Algoritma enkripsi yang mereka buat dikenal dengan nama RSA.

Akhirnya, sejak tahun 1976 berbagai algoritma enkripsi, tanda tangan digital,, pertukaran kunci, dan teknik lain dikembangkan dalam bidang kriptografi kunci publik, misalnya algoritma ElGamal untuk enkripsi dan tanda tangan digital dan algoritma DSA untuk tanda tangan digital. Pada tahun 1980 Neal Koblitz memperkenalkan elliptic-curve cryptography

sebagai keluarga baru yang analog dengan algoritma kriptografi kunci publik. Hingga saat ini kriptografi kunci publik terus berkembang pesat seiring dengan aplikasinya yang luas.

2. Konsep Kriptografi Kunci Publik

Konsep kriptografi kunci publik sederhana dan elegan, tetapi memiliki konsekuensi penggunaan yang hebat. Seperti telah dijelaskan di awal, pada kriptografi kunci publik, setiap pengguna memiliki sepasang kunci, satu kunci untuk enkripsi dan satu kundi untuk dekripsi (Gambar 2); kundi untuk enkripsi diumumkan kepada publik – oleh karena itu tidak rahasia – sehingga dinamakan kunci publik (public key), disimbolkan dengan e. Kunci untuk dekripsi bersifat rahasia sehingga dinamakan kunci privat (private key), disimbolkan dengan d. Karena kunci untuk enkripsi tidak sama dengan kunci dekripsi itulah maka kriptografi kunci publik dinamakan kriptografi nirsimetri.



Gambar 2. Skema kriptografi nirsimetri. Kunci enkripsi (e) tidak sama dengan kunci dekripsi (d). Kunci enkripsi bersifat publik *tidak rahasia), sedangkan kunci dekripsi bersifat privat (rahasia).

Misalkan E adalah fungsi enkripsi dan D adalah fungsi dekripsi. Misalkan (e,d) adalah pasangan kunci untuk enkripsi dan dekripsi sedemikian sehingga

$$Ee(m) = c \text{ dan } Dd(c) = m$$

untuk suatu plainteks m dan chiperteks c. Kedua persamaan ini menyiratkan bahwa dengan mengetrahui e dan c, maka secara komputasi hampir tidak mungkin menemukan m. Asumsi lainnya, dengan mengetahui e, secara komputasi hampir tidak mungkin menurunkan d. Ee digambarkan sebagai fungsi pitu kolong (trapdoor) satu-arah dengan d adalah informasi trapdoor yang diperlukan untuk menghitung fungsi inversinya, D, yang dalam hal ini membuat proses dekripsi dapat dilakukan.

Konsep di atas menjasi penting bila kriptografi kunci publik digunakan untuk mengamankan pertukaran pesan dari dua entitas yang berkomunikasi. Misalkan Alice berkomunikasi dengan Bob. Bob memilih pasangan kunci (e,d). Bob mengirimkan kunci enkripsi e (kunci publik) kepada Alice melalui sembarang saluran tetapi tetap menjaga kerahasiaan kunci dekripsinya, d (kunci privat). Kemudian, Alice mengirim pesan m kepada Bob. Alice mengenkripsikan pesan m menggunakan kunci publik Bob, untuk mendapatkan c = Ee(m), lalu mengirimkan c melalui saluran komunikasi (yang tidak perlu aman). Bob mendekripsi chiperteks menggunakan kunci privatnya untuk memperoleh m = Dd(c).

Skema komunikasi dengan kriptografi kunci publik pada Gambar 3 memperlihatkan perbedaan mendasar sistem asimetri dengan simetri. Bob mengirim kunci publik, e, untuk enkripsi kepada Alice melalui saluran yang tidak perlu aman. Saluran yang tidak perlu aman ini mungkin sama dengan yang digunakan untuk mengirim cipherteks. Carol yang melakukan intersepsi komunikasi mungkin berhasil mendapatkan kunci enkripsi e dan cipherteks c, tetapi karena ia tidak mengetahui kunci dekripsi d, maka ia tidak dapat melakukan dekripsi. Hanya Bob yang mengetahui kunci privatnya sendiri, d, (Alice pun tidak tahu) sehingga ia mendekripsi pesan dari Alice dengan kunci privat tersebut.

Hal yang serupa juga dilakukan jika Bob mengirim pesan kepada Alice. Bob harus mengetahui kunci publik terlebih dahulu sebelum ia mengirim pesan kepada Alice. Bob mengenkripsi pesan dengan kunci publik Alice. Hanya Alice yang mengetahui kunci privatnya sendiri, d, (Bob pun tidak tahu) sehingga ia mendekripsi pesan dari Bob dengan kunci privatnya sendiri (kunci privat Alice).

Dengan sistem kriptografi kunci publik ini Bob dan Alice tidak perlu berbagi kunci yang sama. Baik Bob dan Alice keduanya memiliki sepasang kunci, kunci publik dan kunci privat.

Sistem kriptografi kunci publik yang aman memiliki dua karakteristik sebagai berikut:

1. Komputasi untuk enkripsi/dekripsi pesan mudah dilakukan.
2. Secara komputasi hampir tidak mungkin (infeasible) menurunkan kunci privat, d, bila diketahui kunci publik, e, pasangannya.

Kedua karakteristik di atas dapat dianalogikan dengan dua masalah sebagai berikut:

a. Perkalian vs. pemfaktoran. Mengalikan dua buah bilangan prima, a x b = n mudah, tetapi memfaktorkan n menjadi faktor-faktor primanya lebih sulit.

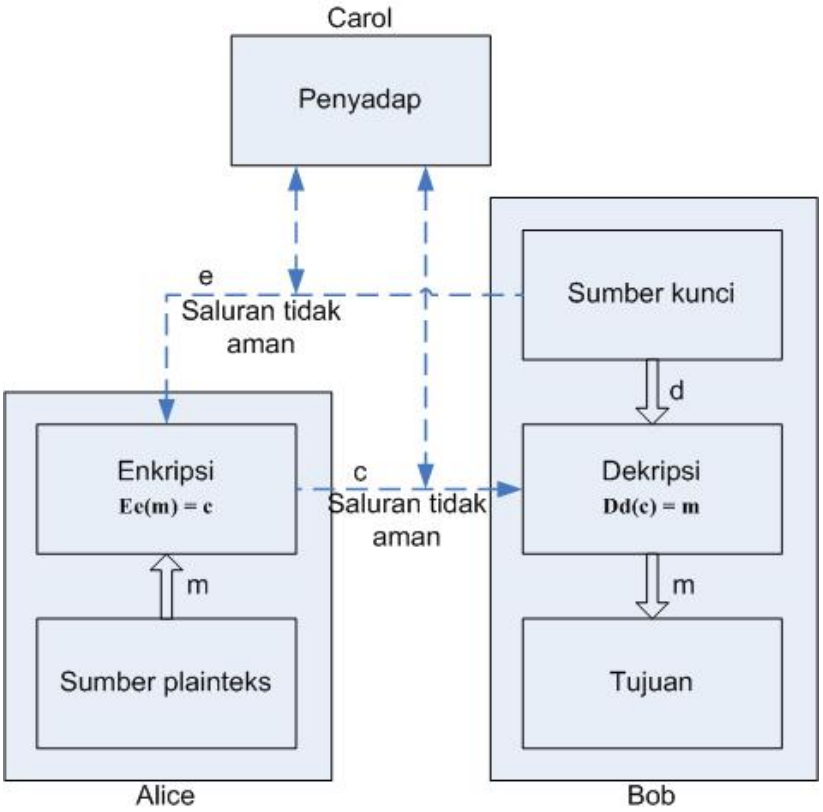
   Contoh:

   31 x 47 = 1457 (perkalian, mudah)

   1457 = ? x ? (pemfaktoran, sulit)

b. Perpangkatan vs. logaritmik diskrit. Melakukan perpangkatan modulo, b = a^x mod n, mudah, tetapi menemukan x dari a^x = b (mod n) lebih sulit.

   Contoh:



Gambar 3. Enkripsi/dekripsi dengan kriptografi kunci publik

   12^6 mod 1125 = 234 (perpangkatan modulo, mudah)

   Carilah x dari $12^x \equiv 234$ (mod 1125) (logaritmik diskrit, sulit)

Dua masalah matematika di atas sering dijadikan dasar pembangkitan sepasang kunci pada kriptografi kunci publik, yaitu:

1. Pemfaktoran

   Diberikan bilangan bulat n. Faktorkan menjadi faktor perimanya

   Contoh:

   10 = 2x5

   60 = 2 x 2 x 3 x 5

   252601 = 41 x 61 x 101

   213 – 1 = 3391 x 23279 x 65993 x 1868569 x 1066818132868207

   Semakin besar n, semakin sulit memfaktorkan (butuh waktu yang sangat lama). Algoritma yang menggunakan prinsip ini: RSA.

2. Logaritma diskrit

   Temukan x sedemikian sehingga $a^x \equiv b \pmod n$ sulit dihitung.

   Contoh: jika $3^x \equiv 15 \pmod{17}$ maka x = 6.

   Semakin besar a, b, n semakin sulit memfaktorkan (butuh waktu yang lama). Algoritma yang menggunakan prinsip ini: ElGamal dan DSA.

Perbandingan Kriptografi Kunci Simetri dengan kriptografi Kunci Publik

Baik kriptografi kunci simetri maupun kriptografi asimetri (kunci publik), keduanya memiliki kelebihan dan kelemahan.

Kelebihan kriptografi kunci simetri:

1. Algoritma kriptografi simetri dirancang sehingga proses enkripsi/dekripsi memiliki waktu yang singkat.
2. ukuran kunci simetri relatif pendek. Algoritma kriptografi kunci simetri dapat digunakan untuk membangkitkan bilangan acak.
3. Algoritma kriptografi simetri dapat disusun untuk menghasilkan cipher yang lebih kuat.
4. Otentikasi pengirim pesan langsung diketahui dari cipherteks yang diterima, karena kunci hanya diketahui oleh pengirim dan penerima pesan saja.

Kelemahan kriptografi kunci simetri:

1. Kunci rahasia harus dikirim melalui saluran yang aman. Kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci ini.
2. Kunci harus sering diubah, mungkin pada setiap sesi komunikasi.

Kelebihan kriptografi kunci publik (nirsimetri):

1. Hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi (tetapi, otentikasi kunci publik tetap harus terjamin). Tidak ada kebutuhan mengirim kunci privat sebagaimana pada sistem simetri.
2. Pasangan kunci publik/kunci privat tidak perlu diubah, bahkan dalam periode waktu yang panjang.
3. Dapat digunakan untuk mengamankan pengiriman kunci simetri.
4. Beberapa algoritma kriptografi kunci publik dapat digunakan untuk memberi tanda tangan digital pada pesan.

Kelemahan kriptografi kunci publik (nirsimetri):

1. Enkripsi dan dekripsi data pada umumnya lebih lambat daripada sistem simetri, karena enkripsi dan dekripsi menggunakan bilangan yang besar dan melibatkan operasi perpangkatan yang besar.
2. Ukuran cipherteks lebih besar daripada plainteks (bisa dua hingga empat kali ukuran plainteks).
3. Ukuran kunci relatif lebih besar daripad ukuran kunci simetri.
4. karena kunci publik diketahui secara luas dan dapat digunakan setiap orang, maka cipherteks tidak memberikan informasi mengenai otentikasi pengirim.
5. Tidak ada algoritma kunci publik yang terbukti aman (sama seperti block cipher). Kebanyakan algoritma mendasarkan keamanannya pada sulitnya memecahkan persoalan-persoalan aritmetik (pemfaktoran, logaritmik, dan lain-lain) yang menjadi dasar pembangkitan kunci. Kriptografi kunci publik juga tidak aman dari serangan man-in-the-middle-attack. Orang di "tengah" mengintersepsi komunikasi lalu berpura-pura sebagai salah satu pihak yang berkomunikasi untuk mengetahui informasi rahasia.

Karena kelebihan dan kekurangan yang ada dalam masing-masing kriptografi, maka tidaklah benar bahwa kriptografi kunci publik menggantikan kriptografi kunci simetri. Karena kriptografi kunci publik mempunyai kelemahan dari segi komputasi dan ukuran cipherteks dibandingkan dengan kriptografi kunci simetri, maka hal ini mampnyai implikasi dalam praktek penggunaannya. Kebanyakan sistem keamanan menggunakan gabungan kriptografi kunci simetri dan kriptografi kunci publik (hybrid cryptosystem). Pada sistem hibrida ini, enkripsi/dekripsi pesan dilakukan menggunakan kriptografi kunci simetri, sedangkan kunci simetri dienkripsi/dekripsi menggunakan kriptografi kunci publik. Kunci simetri (yang juga disebut

kunci sesi) dibangkitkan oleh salah satu pihak den mengenkripsi pesan dengan kunci tersebut. Selanjutnya kunci sesi dienkripsi dengan kunci publik penerima lelu dikirim bersama-sama dengan pesan yang sudah dienkripsi. Penerima mula-mula mendekripsi kunci sesi dengan kunci privatnya, lalu mendekripsi pesan dengan kunci sesi tersebut.

3. Aplikasi Kriptografi Kunci Publik

Aplikasi kriptografi kunci publik dibagi menjadi 3 kategori:

1. Kerahasiaan data

   Seperti pada kriptografi kunci simetri, kriptografi kunci publik dapat digunakan untuk menjaga kerahasiaan data (provide confidentiality/secrecy) melalui mekanisme enkripsi dan dekripsi. Contoh algoritma untuk aplikasi ini adalah RSA, Knapsack, Rabin, ElGamal, Elliptic Curve Cryptography (ECC).

2. Tanda tangan digital

   Tanda tangan digital (digital signature) dengan menggunakan algoritma kriptografi kunci publik digunakan untuk membuktikan otentikasi pesan maupun otentikasi pengirim (provide authentication). Contoh algoritmanya untuk aplikasi ini adalah RSA, DSA dan ElGamal.

3. Pertukaran kunci (key exchange)

   Algoritma kriptografi kunci publik dapat digunakan untuk pengirima kunci simetri (session keys). Contoh algoritmanya adalah RSA dan Diffie-Hellman.

Beberapa algoritma kriptografi kunci publik dapat digunakan untuk ketiga macam kategori aplikasi (misalnya RSA), beberapa algoritma hanya ditujukan untuk aplikasi spesifik (misalnya DSA untuk digital signature).

RSA

Dari sekian banyak algoritma kunci publik yang pernah dibuat, algoritma yang paling populer adalah algoritma RSA. Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (Massachusetts Institute of Technology) pada tahun 1976, yaitu: Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci privat. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin.

Algoritma RSA memiliki besaran-besaran sebagai berikut:

1. p dan q, bilangan prima (rahasia)
2. n = p . q (tidak rahasia)
3. $\Phi(n) = (p-1)(q-1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plainteks) (rahasia)
7. c (chiperteks) (tidak rahasia)

Perumusan Algoritma RSA

Algoritma RSA didasarkan pada teorema Euler yang menyatakan bahwa

$$a^{\Phi(n)} \equiv 1 \pmod{n} \qquad (1)$$

dengan syarat:

1. a harus relatif prima terhadap n
2. $\Phi(n) = n(1 - 1/p1)(1 - 1/p2)\ldots(1 - 1/pr)$, yang dalam hal ini p1, p2, p3,…,pr adalah faktor prima dari n. $\Phi(n)$ adalah fungsi yang menentukan berapa banyak dari bilangan-bilangan 1, 2, 3, …, n yang relatif prima terhadap n.

Berdasarkan sifat $ak \equiv bk \pmod{n}$ untuk k nilangan bulat ≥ 1, maka persamaan (1) di atas dapat ditulis menjadi

$$a^{k\Phi(n)} \equiv 1^k \pmod{n} \qquad (2)$$

atau

$$a^{k\Phi(n)} \equiv 1 \pmod{n} \qquad (3)$$

Bila a diganti dengan m, maka persamaan (3) dapat ditulis menjadi

$$m^{k\Phi(n)} \equiv 1^k \pmod{n} \qquad (4)$$

Berdasarkan sifat ac ≡ bc (mod n) maka bila persamaan (4) dikalikan dengan m menjadi

$$m^{k\Phi(n)+1} \equiv m \pmod{n} \qquad (5)$$

yang dalam hal ini relatif prima terhadap n.

Misalkan e dan d dipilih sedemikian sehingga

$$e \cdot d \equiv 1 \pmod{\Phi(n)} \qquad (6)$$

atau

$$e \cdot d \equiv k\Phi(n) + 1 \qquad (7)$$

Sulihkan persamaan (7) ke dalam persamaan (5) menjadi

$$m^{e \cdot d} \equiv m \pmod{n} \qquad (8)$$

Persamaan (8) dapat ditulis kembali menjadi

$$(m^e)^d \equiv m \pmod{n} \qquad (9)$$

yang artinya, perpangkatan m dengan e diikuti dengan perpangkatan dengan d menghasilkan kembali m semula. Berdasarkan persamaan (9), maka enkripsi dan dekripsi dirumuskan sebagai berikut:

$$E_e(m) \equiv c \equiv m^e \bmod n \qquad (10)$$

$$D_d(c) \equiv m \equiv c^d \bmod n \qquad (11)$$

Karena e . d = d . e, maka enkripsi diikuti dengan dekripsi ekivalen dengan dekripsi diikuti enkripsi:

$$D_d(E_e(m)) = E_e(D_d(m)) = m^d \bmod n \qquad (12)$$

Oleh karena $m^d \bmod n \equiv (m + jn)^d \bmod n$ untuk sembarang bilangan bulat j, maka tiap plainteks m, m +n, m + 2n, …, menghasilkan cipher yang sama. Dengan kata lain, transformasinya dari banyak ke satu. Agar transformasinya satu ke satu, maka m harus dibatasi dalam himpunan {0, 1, 2, …, n-1} sehingga enkripsi dan dekripsi tetap benar seperti dalam persamaan (10) dan (11).

Algoritma Membangkitkan Pasangan Kunci

1. Pilih dua buah bilangan prima sembarang, p dan q.
2. Hitung n = p . q (sebaiknya $p \neq q$, sebab jika p = q maka $n = p^2$ sehingga p dapat diperoleh dengan menarik akar pangkat dua dari n).
3. Hitung Φ(n) = (p-1)(q-1).
4. Pilih kunci publik, e, yang relatif prima terhadap Φ(n).
5. Bangkitkan kunci privat dengan menggunakan persamaan (6), yaitu e . d ≡ 1 (mod Φ(n)).

   Perhatikan bahwa e . d ≡ 1 (mod Φ(n)) ekivalen dengan e . d = 1 + kΦ(n), sehingga secara sederhana d dapat dihitung dengan

$$d = \frac{1 + k\Phi(n)}{e} \qquad (13)$$

Hasil dari algoritma di atas adalah:

1. Kunci publik adalah pasangan (e,n)
2. Kunci privat adalah pasangan (d,n)

N tidak bersifat rahasia, sebab ia diperlukan pada perhitungan enkripsi/dekripsi.

Algoritma Enkripsi/Dekripsi

Enkripsi:

1. Ambil kunci publik penerima pesan, e, dan modulus n.
2. Nyatakan plainteks m menjadi blok-blok m1, m2, …, sedemikian sehingga setiap blok merepresentasikan nilai di dalam selang [0, n-1].
3. Setiap blok m_i dienkripsi menjadi blok c_i denagn rumus $c_i = m_i^e \bmod n$.

Dekripsi:

1. Setiap blok chiperteks c_i didekripsi kembali menjadi blok m_i dengan rumus $m_i = c_i^d \bmod n$.

Keamanan RSA

Keamanan algoritma RSA didasarkan pada sulitnya memfaktorkan bilangan besar menjadi faktor-faktor primanya.

> *Masalah pemfaktoran*: Faktorkan n, yang dalam hal ini n adalah hasil kali dari dua atau lebih bilangan prima.

Pada RSA, masalah pemfaktoran berbunyi: Faktorkan n menjadi dua faktor primanya, p dan q, sedemikian sehingga n = p . q. Sekali n berhasil difaktorkan menjadi p dan q, maka $\Phi(n) = (p-1)(q-1)$ dapat dihitung. Selanjutnya, karena kunci enkripsi e diumumkan (tidak rahasia), maka kunci dekripsi d dapat dihitung dari persamaan $e \cdot d \equiv 1 \pmod{\Phi(n)}$.

Selama 300 tahun para matematikawan mencoba mencari faktor bilangan yang besar namun tidak banyak membuahkan hasil. Semua buktui yang diketahui menunjukkan bahwa upaya pemfaktoran itu luar biasa sulit. Belum ditemukan algoritma pemfaktoran bilangan besar dalam waktu polinomial, tetapi juga tidak dapat dibuktikan algoritma tersebut ada. Fakta inilah yang membuat algoritma RSA dianggap aman. Penemu algoritma RSA bahkan menyarankan nilai p dan q panjangnya lebih dari 100 angka. Dengan demikian hasil kali n = p x q akan berukuran lebih besar dari 200 angka. Menurut Rivest dan kawan-kawan, usaha untuk mencari faktor prima dari bilangan 200 angka membutuhkan waktu komputasi selama 4 milyar tahun, sedangkan untuk bilangan 500 angka membutuhkan waktu 10^{25} tahun (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik).

Secara umum dapat disimpulkan bahwa RSA hanya aman jika n cukup besar. Jika panjang n hanya 256 bit saja atau kurang, ia dapat difaktorkan dalam beberapa jam saja dengan sebuah komputer PC dan program yang tersedia secara bebas. Jika panjang n 512 bit atau kurang, ia dapat difaktorkan dengan beberapa ratus komputer.

Tahun 1977, tiga orang penemu RSA membuat sayembara untuk memecahkan cipherteks dengan menggunakan RSA di majalah Scientific American. Hadiahnya adalah $100. Tahun 1994, kelompok yang bekerja dengan kolaborasi internet berhasil memecahkan cipherteks dalam waktu 8 bulan.

Kecepatan

RSA lebih lambat daripada algoritma kriptografi kunci simetri seperti AES dan DES. Dalam praktek, pesan dienkripsi dengan kunci rahasia dengan menggunakan salah satu algoritma kunci simetri, sedangkan RSA digunakan untuk mengenkripsi kunci rahasia. Pesan dan kunci rahasia yang masing-masing sudah dienkripsi dikirim bersama-sama. Penerima pesan mula-mula mendekripsi kunci rahasia dengan kunci privatnya, lalu menggunakan kunci rahasia tersebut untuk mendekripsi pesan.

Man-In-The-Middle Attack

Karena pengirim dan penerima harus berbagi kunci publik, maka distribusi kunci publik dapat mengalami serangan *Man-In-The-Middle Attack*. Misalkan Alice dan Bob mengirim kunci publiknya masing-masing melalui saluran komunikasi. Orang di tengah, misalkan Carol, memutus komunikasi antara Bob dan Alice lalu ia berpura-pura sebagai salah satu pihak (Alice atau Bob). Carol (yang menyamar sebagai Alice) mengirimkan kunci publiknya kepada Bob (Bob percaya itu adalah kunci publik Alice), dan Carol (yang menyamar sebagai Bob) mengirimkan kunci publiknya kepada Alice (Alice percaya bahwa itu adalah kunci publik Bob). Selanjutnya, Carol mendekripsi pesan dari Bob dengan kunci privatnya, menyimpan salinannya, lalu mengenkripsi pesan tersebut dengan kunci publik Alice, dan mengirim cipherteks tersebut kepada Alice. Alice dan Bob tidak dapat mendeteksi keberadaan Carol.

Chosen-Plaintext Attack

RSA mudah diserang dengan chosen-plaintext attack. Misalkan kriptanalis memiliki beberapa plainteks dari pesan. Ia dapat memilih beberapa plainteks untuk dienkripsi denga kunci publik, lalu menyimpan hasilnya di dalam kamus. Kemudian kriptanalis menyadap saluran komunikasi dan membandingkan cipherteks yeng disadap dengan cipherteks di dalam kamus. Jika terdapat kesamaan, maka kriptanalis dapat menggunakan kamus tersebut untuk mempelajari isi pesan.

4. Kriptografi Kunci Publik Di Dunia Nyata

Sistem kriptografi kunci publik cocok untuk digunakan di dalam kelompok pengguna di lingkungan jaringan komputer (LAN/WAN) yang memungkinkan mereka saling berkomunikasi. Setiap pengguna jaringan mempunyai pasangan kunci publik dan kunci privat yang bersesuaian. Kunci publik, karena tidak rahasia, biasanya disimpan di dalam basisdata kunci yang dapat diakses oleh pengguna lain. Jika ada pengguna yang hendak berkirim pesan ke pengguna lainnya, maka ia perlu mengetahui kunci publik penerima pesan melalui basisdata kunci ini lalu menggunakannya untuk mengenkripsi pesan. Hanya penerima pesan saja yang dapat mendekripsi pesan karena ia yang mengetahui kunci privatnya sendiri.

Dengan sistem kriptografi kunci publik, tidak diperlukan pengiriman kunci privat melalui saluran komunikasi khusus sebagaimana pada sistem kriptografi simetri. Meskipun kunci publik diumumkan ke setiap orang di dalam kelompok, namun kunci publik perlu dilindungi agar otentikasinya terjamin (misalnya tidak dimanipulasi oleh orang lain).

Dalam jaringan yang sangat luas (misalnya internet), semua hal yang membutuhkan pembuatan kunci atau perubahan kunci publik dapat memakan waktu cukup lama untuk diinformasikan kepada semua pihak yang perlu diinformasikan. Untuk alasan ini, sistem yang harus bereaksi terhadap kejadian secara *real-time* (misalnya sistem yang sangat penting atau sistem keamanan nasional) harus menggunakan kriptografi kunci publik dengan penuh kehati-hatian.

Hak Khusus Pembangkitan Kunci

Pembangkitan kunci yang salah di dalam suatu sistem dapat menyebabkan kesalahan total dalam sistem. Jika kunci publik dapat dibangkitkan oleh setiap pengguna, kemungkinan tersebut dapat terjadi. Namun, ada beberapa pendekatan perancangan yang dapat mengurangi kemungkinan buruk ini terjadi. Contohnya, dengan pemberian hak kepada satu atau beberapa pihak untuk membangkitkan kunci publik. Misalnya hak tersebut diberikan kepada Alice dan Bob. Kini hanya keduanya yang dapat membuat kunci publik, dan keduanya harus melaksanakannya bersama-sama.

Pembangkitan Kunci Baru

Setelah sebuah kunci dibangkitkan, atau pengguna baru masuk ke dalm sistem, kunci publik baru yang bersangkutan harus didistribusikan dengan cara-cara yang telah diatur.

Misalnya kunci publik milik Carol baru saja dibangkikan kembali (misalnya karena kunci lama telah kedaluarsa). Hingga kunci baru milik Carol telah didistribusikan ke semua pihak dalam sistem, Carol diputus kontakna dari sistem. Tak ada orang di dalam sistem yang dapat mengirim pesan kepada Carol tanpa melanggar protokol komunikasi (misalnya mengirim pesan tanpa dienkripsi), dan pesan dari Carol tidak dapat dikirimkan dengan enkripsi terlebih dahulu karena alasan yang sama.

Penyebaran Informasi Pembangkitan Kunci

Notifikasi atas pembangkitan sertifikat kuncu harus disebarkan kepada semua pemegang potensialnya (semua yang akan membutuhkannya), dengan sesegera mungkin.

Ada dua cara penyebaran informasi (dalam hal ini penyebaran informasi pembangkitan kunci) dalam suatu sistem terdistribusi:

1. Informasi diberikan dari pusat kepada pengguna jaringan (push).
2. Pengguna menarik informasi dari pusat (pull).

Pemulihan Dari Kebocoran Kunci

Asumsikan bahwa pemegang otoritas pembangkitan kunci memutuskan bahwa kunci tertentu harus dibangkitkan ulang. Dalam kasus terbanyak, hal ini terjadi karena diketahui bahwa beberapa kejadian di masa lalu membahayakan kerahasiaan kunci privat. Misalkan waktu ini diasumsikan sebagai T.

Ada dua hal yang menjadi implikasi dari masalah di atas. Pesan yang dienkripsi dengan kunci publik (sekarang atau dulu) tidak dapat lagi dianggap benar-benar aman. Kedua, tanda tangan digital yang dibuat dengan kunci privat yang bersangkutan (yang dianggap tidak aman) setelah waktu T, tidak dapat lagi dianggap otentik tanpa informasi mengenai siapa, kapan, di mana, dan lain-lain yang berkaitan dengan kejadian pembangkitan tanda tangan digital tersebut.

Prosedur pemulihan ini dapat menjadi kompleks, dan dalam prosesnya sistem dapat menjadi lebih rantan terhadap serangan *Denial of Service*.

5. Kesimpulan

Kriptografi kunci publik memiliki berbagai kelebihan dari kriptografi kunci simetri dalam segi keamanan tapi juga memiliki kekurangan dalam hal efisiensi penggunaannya. Oleh karena itu, upaya penggabungan keduanya dilakukan untuk memperoleh hasil yang optimal dari kedua segi, keamanan dan efisiensi.

Kegunaannya yang luas menyebabkan kriptografi kunci publik banyak dipakai dalam banyak aplikasi, khususnya yang berkaitan dengan otentikasi data. Salah satu algoritmanya yang sukses, RSA kini banyak dipakai di jaringan di seluruh dunia. Ini membuktikan bahwa teknologi kriptografi ini masih dapat dipercaya dengan baik oleh kalangan penggunanya.

Namun, dalam jaringan yang luas, ada beberapa hal yang harus diperhatikan mengenai kriptografi kunci publik terkait penggunaannya oleh banyak pihak. Perlu ada sistem tertentu yang manengani pembuatan kunci pengguna agar tidak terjadi kesemrawutan. Selain itu hal-lain yang perlu diperhatikan adalah mengenai pembangkitan kunci untuk suatu pihak, informasi mengenai pembangkitan tersebut, serta pemulihan dari kebocoran kunci. Hal-hal tersebut perlu diperhatikan agar semua pihak yang terkait dengan sistem kriptografi kunci publik terjamin keamanannya.

6. Pustaka

Kriptografi. Munir, Rinaldi. 2006

***http://en.wikipedia.org**. Desember-Januari 2006.*