APLIKASI TEORI BILANGAN DALAM KRIPTOGRAFI KUNCI PUBLIK DAN ALGORITMA DIFFIE-HELLMAN

Aswin Juari – NIM : 13505076
Departemen Teknik Informatika, Institut teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if15076@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang aplikasi matematika diskret dalam bidang Kriptografi dan Algoritma Pertukaran Kunci Diffie-Hellman (*Diffie-Hellman Key Exchange*). Secara umum, semua kriptografi Kunci-Publik berdasarkan pengolahan fungsi matematika. Kriptografi kunci-publik adalah sebuah ilmu penjagaan kerahasiaan pesan dengan menggunakan dua buah kunci yang tidak sama, satu kunci dijaga privat, dan satu kunci yang lain diumumkan ke publik.

Kriptogafi kunci-publik berguna dalam hal enkripsi/dekripsi, *Digital Signature*, dan pertukaran kunci. Namun, tidak semua algoritma dalam Kriptografi kunci-pubik berguna untuk semua tujuan. Ada keterbatasan tujuan yang dimiliki oleh setiap algoritma. Selain itu, kriptografi kunci-publik memiliki beberapa kelemahan.

Algoritma Diffie-Hellman atau Petukaran Kunci Diffie-Hellman merupakan salah satu aplikasi dari kriptografi kunci-publik, di mana algoritma ini menggunakan fungsi aritmatika modulo dalam pembangkitan kunci rahasia. Tujuan utama dari algoritma ini adalah membuat pengguna dapat melakukan "pertukaran" kunci secara aman karena kunci rahasia yang dibangkitkan oleh A dan B adalah sama tetapi nilai privat yang dimiliki oleh A dan B tidak sama. Algoritma ini terbatas pada pertukaran kunci saja.

Kata kunci: *Plaintext, Ciphertext, encryption, decryption, Digital signature, cryptanalys, authenticator, Key exchange, trap-door one-way function, private value, primitive root, discrete logarithm*

1. Pendahuluan

Perkembangan Kriptogafi Kunci Publik adalah yang paling besar di sepanjang sejarah kriptografi. Dari yang paling awal hingga modern, secara virtual semua sistem kriptografi telah berdasarkan pada kakas substitusi dan permutasi sederhana. Setelah milenia bekerja dengan algoritma yang secara dasarnya dapat dihitung oleh tangan, perkembangan uatama dalam kriptogafi simetris terjadi dengan perkembangan mesin enkripsi/dekripsi rotor. Rotor elektromekanik membuat perkembangan sistem cipher yang lebih kompleks. Dengan adanya komputer, bahkan ditemukannya sistem yang lebih kompleks, yang paling mencolok adalah *Data Encryption Standard* (DES). Akan tetapi, baik mesin rotor maupun DES, walaupun menunjukan perkembangan signifikan, kedua-duanya bergantung pada kakas substitusi dan permutasi.

Kriptografi Kunci-Publik membuat perubahan radikal terhadap yang sudah ada. Dalam satu hal, algoritma kunci-publik berdasarkan pada fungsi-fungsi matematika dibandingkan dengan permutasi dan kombinasi. Hal yang lebih penting lagi, kriptografi kunci-publik tidak simetris, melibatkan penggunaan 2 jenis kunci, kontras dengan enkripsi simetris, yang hanya menggunakan satu jenis kunci. Penggunaan dua kunci telah membuat konsekuensi pada bidang kerahasiaan, distribusi kunci, dan autentikasi.

Sebelum melanjutkan ke pokok bahasan utama, kita akan menyebutkan beberapa kesalahan konsep mengenai enkripsi kunci-publik. Kesalahan pertama adalah enkripsi kunci-publik lebih aman dari pemecahan pembacaan sandi daripada enkripsi simetris. Sebenarnya, tingkat keamanan dari sebuah skema enkripsi bergantung pada panjang kunci dan kerja komputasi yang terlibat dalam pemecahan cipher. Tidak ada dasar terhadap baik enkripsi simetris maupun enkripsi kunci-publik yang membuat salah satu lebih baik daripada yang lain dari sudut pandang ketahanan terhadap pemecahan pembacaan sandi.

Kesalahan konsep kedua adalah bahwa enkripsi kunci-publik adalah teknik untuk tujuan umum sehingga membuat enkripsi simetri tidak terpakai lagi. Sebaliknya, karena komputasi di atas dari skema enkripsi kunci-publik, tampaknya bahwa tidak ada kemungkinan yang dapat diduga bahwa enkripsi simetris akan ditinggalkan.

Terakhir, ada perasaan bahwa distribusi kunci adalah bukan hal yang penting ketika menggunakan enkripsi kunci-publik. Sebenarnya, beberapa bentuk protokol tetap dibutuhkan umumnya melibatkan agen central, dan prosedur yang dilibatkan tidak lebih mudah ataupun lebih efisien dibandingkan dengan enkripsi simetris.

Beberapa contoh dari kriptografi Kunci-Publik yang terkenal luas adalah Diffie-Hellman, DSS (*Digital Signature Standard*), ElGamal, Elliptic Curve, dan lain-lain.

2. Dasar-dasar Kriptosistem Kunci Publik

2. 1. Kriptosistem Kunci-Publik

Algoritma kunci-publik bergantung pada satu kunci enkripsi dan kunci dekripsi yang berbeda namun tetap berhubungan. Algoritma ini memiliki karakteristik penting seperti berikut:

- Secara komputasional sulit untuk menentukan kunci dekripsi yang hanya diberikan pengetahuan algoritma kripsografi dan kunci dekripsinya.

Selain itu, dalam beeberapa algoritma, seperti RSA, juga menunjukkan sifat-sifat berikut:

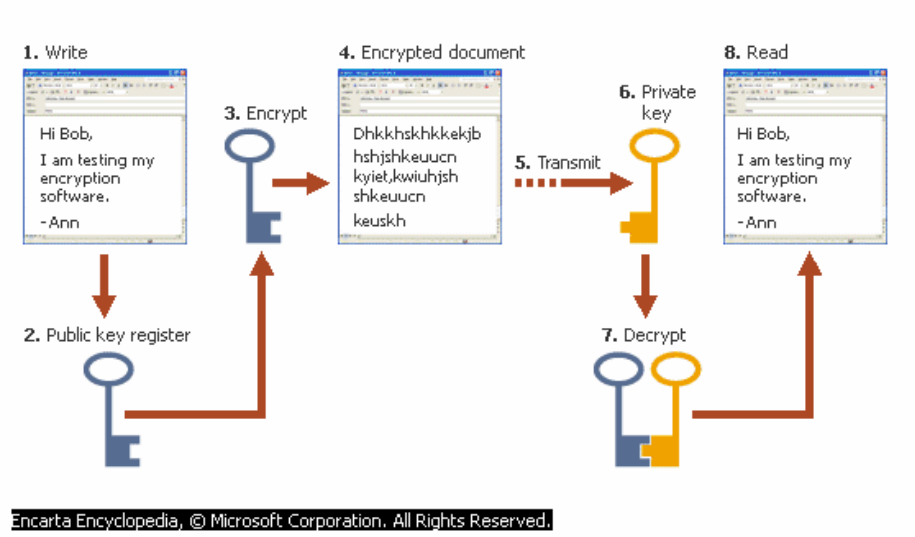
- Salah satu dari dua kunci yang saling berhubungan dapat digunakan untuk enkripsi, sementara yang lain digunakan untuk dekripsi.

Skema enkripsi kunci-publik memiliki 6 hal:

- ***Plaintext*:** data atau pesan yang dapat dibaca sebagai input dari algoritma.
- ***Encryption algorithm*:** algoritma enkripsi yang melakukan berbagai transformasi pada *Plaintext*
- ***Public and private key*:** Ini adalah pasangan kunci yang telah dipilih sehingga jika salah satu digunakan untuk enkripsi, maka yang lain digunakan sebagai dekripsi. Transformasi eksak dilakukan oleh algoritma enkripsi yang bergantung pada kunci publik atau pribadi yang digunakan sebgai input.
- ***Chipertext*:** Ini adalah pesan yang telah dikacaukan yang diproduksi sebagai output. Untuk satu pesan, dua buah kunci yang berbeda akan menghasilkan dua buah ciphertext yang berbeda.
- ***Decryption algorithm*:** Algoritma ini menerima ciphertext dan kunci yang sesuai dan menhasilakn *plaintext* awal.

Langkah-langkah dasarnya sebagai berikut:

1. Masing-masing *user* membangkitkan pasangan kunci yang akan digunakan untuk enkripsi dan dekripsi.
2. Masing-masing *user* menempatkan salah satu dari dua kunci pada sebuah register publik atau pada file yang dapat diakses lain. Ini adalah kunci publik. Satu kunci lagi dijaga privat.
3. Jika Ann ingin mengirimkan pesan rahasia terhadap Bob, maka Ann mengenkripsikan pesan tersebut menggunakan kunci-publik Bob.
4. Ketika Bob membaca pesan, dia mendekripsi pesan tersebut menggunakan kunci privatnya. Tidak ada penerima lain yang dapat mendekripsi karena hanya dia sendiri yang mengetahui kunci privatnya.



Gambar 1:*Encryption*

Dengan pendekatan ini, semua orang dapat mengakses kunci publik dan kunci privat dibangkitkan secara lokal oleh setiap peserta. Oleh karena itu, kunci privat tidak perlu didistribusikan. Selama sistem mejaga kunci privat, komunikasi yang terjadi tetap aman. Pada suatu waktu, sistem dapat mengubah kunci privat dan menerbitkan kunci publik baru yang sesuai untuk menggantikan kunci publik yang lama.

Tabel 1 merangkum beberapa aspek penting pada enkripsi kunci-publik dan enkripsi simetris. Untuk membedakan keduanya, kita akan secara umum merujuk pada kunci yang digunakan pada enkripsi simteris sebagai kunci rahasia. Dua kunci yang digunakan pada enkripsi kunci-publik dinyatakan sebagai kunci-publik dan kunci privat. Kunci privat selalu dijaga kerahasiaannya, tetapi dirujuk sebgai kunci privat daripada kunci rahasia untuk menghindari kebingungan dengan enkripsi simetris.

Conventional Encryption	***Public-key Encryption***
Yang harus bekerja: 1. Algoritma yang sama dengan kunci yang sama digunakan untuk enkripsi maupun dekripsi. 2. Penerima dan pemberi harus membagikan algoritma dan kunci. Yang diperlukan untuk keamanan: 1. Kunci harus dijaga kerahasiaannya. 2. Sangatlah tidak mungkin atau setidaknya tidak praktis untuk menguraikan sebuah pesan jika tidak ada informasi yang tersedia. 3. Pengetahuan mengenai algoritma dan sampel *chipertext* pasti tidak cukup untuk menentukan kunci.	Yang harus bekerja: 1. Satu algoritma yang digunakan untuk enkripsi dan dekripsi dengan satu pasang kunci, satu untuk enkripsi dan satunya lagi untuk dekripsi. 2. Pengirim dan penerima masing-masing harus memiliki satu dari pasangan kunci yang bersesuaian (bukan kunci yang sama). Yang diperlukan untuk keamanan: 1. Salah satu dari dua buah kunci harus dijaga kerahasiaannya. 2. Sangatlah tidak mungkin atau setidaknya tidak praktis untuk menguraikan sebuah pesan jika tidak ada informasi yang tersedia. 3. Pengetahuan mengenai algoritma, salah satu kunci, dan sampel *chipertext* pasti tidak cukup untuk menentukan kunci.

Tabel 1

Mari kita melihat lebih dekat pada elemen-elemen dasar pada skema enkripsi kunci publik, menggunakan Tabel 1. Ada beberapa sumber A yang memproduksi pesan dalam *plaintext*, $X = [X_1, X_2, \ldots X_M]$. M elemen dari X adalah huruf dalam beberapa alphabet terbatas. Pesan ditujukan pada B. B membangkitkan pasangan kunci yang berhubungan; kunci publik, KU_b, dan sebuah kunci privat, KR_B. KR_B hanya diketahui oleh B, sementara KU_b diketahui publik dan oleh karena itu dapat diakses oleh A.

Dengan pesan X dan kunci enkripsi KU_b sebagai input, A membentuk chipertext $Y = [Y_1, Y_2, \ldots, Y_N]$:

$$Y = E_{KUb}(X)$$

Penerima yang dimaksudkan, yang memiliki kunci privat, mampu membalikkan tansformasi:

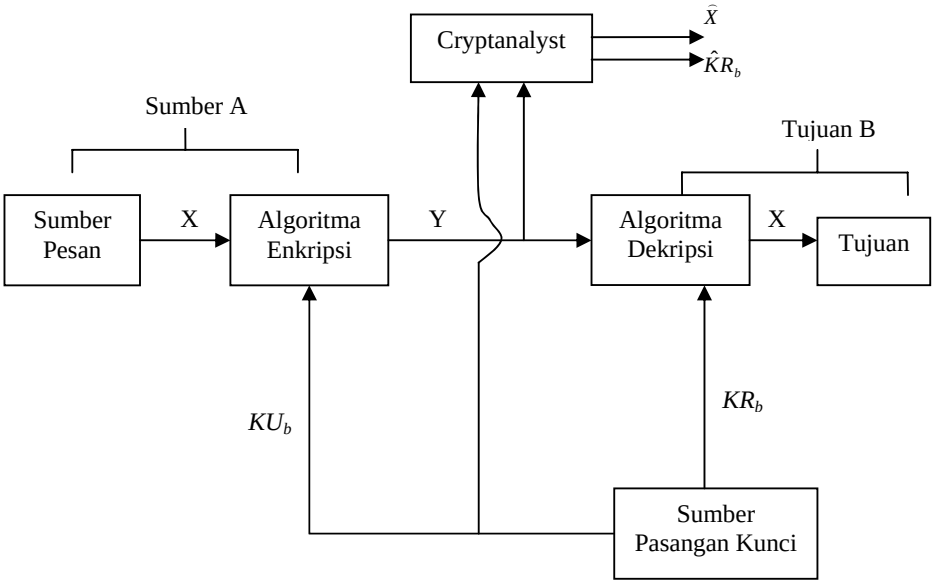
$$X = D_{KR}(Y)$$

Musuh, mengamati Y dan memiliki akses terhadap KU_b, tetapi tak memiliki akses terhadap KR_b atau X, harus mencoba untuk mengembalikan X dan/atau KR_b. Diasumsikan bahwa musuh memiliki pengetahuan tentang algoritma enkripsi (E) dan dekripsi (D). Jika musuh hanya tertarik pada pesan tertentu ini saja, maka fokus usaha adalah untuk mengembalikan X, dengan membangkitkan *plaintext* perkiraan $\hat{X}$. Walaupun begitu, seringkali musuh tertarik untuk dapat membaca pesan yang akan datang pula, di mana pada kasus ini adalah berusaha mendapatkan KR_b dengan membangkitkan perkiraan $\hat{KR}_b$.

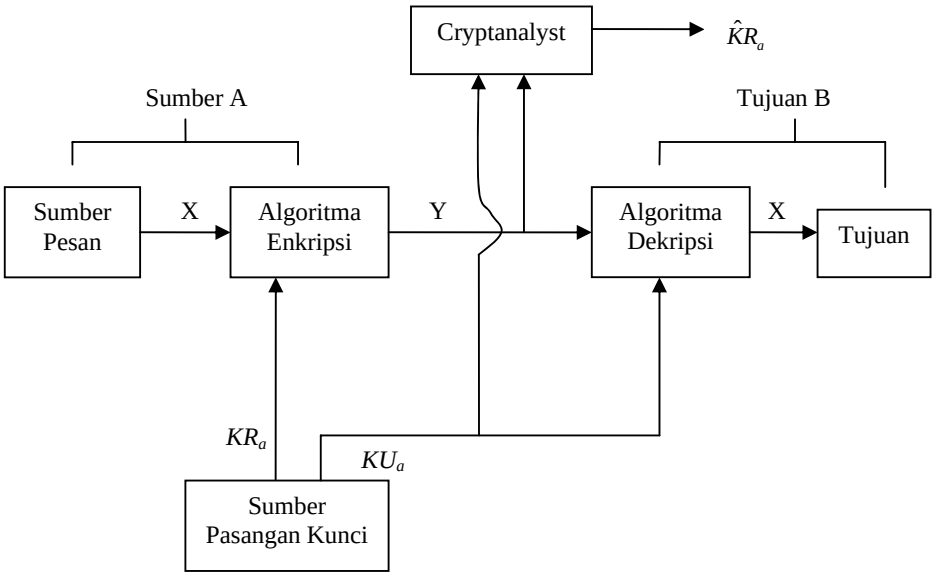
Kita telah menyebutkansebelumnya bahwa salah satu dari dua kunci yang saling berhubungan dapat digunakan untuk enkripsi, dan yang lain digunakan sebagai dekripsi. Hal ini membuat sekma kriptografik yang agak berbeda diterapkan. Skema diilustrasikan pada gambar 2 yang menerangkan kerahasiaan dan gambar 3 yang menerangkan autentikasi:

$$Y = E_{KRa}(X)$$
$$X = D_{KUa}(Y)$$

Dalam kasus ini, A menyiapkan sebuah pesan kepada B dan mengenkripsinya dengan menggunakan kunci privat A sebelum mengirimkannya. B dapat mendekripsi pesan tersebut dengan menggunakan kunci publik A. Karena pesan tersebut dienkripsi dengan menggunakan kunci privat A, hanya A yang dapat menyiapkan pesan tersebut. Jadi, keseluruhan pesan yang dienkripsi menjadi *digital signature*. Tambahan lagi, tidak mungkin mengubah pesan tersebut tanpa mengakses kunci privat A, jadi pesan itu diautentikasi baik dalam hal sumber maupun integritas data.



Gambar 2: Kerahasiaan



Gambar 3 :Autentikasi

Dalam skema terdahulu, semua pesan dienkripsi, yang , walaupun memvalidasi baik penulis pesan, maupun isi pesan, menggunakan jumlah penyimpanan yang besar. Masing-masing dokumen harus dijaga dalam bentuk *plaintext* untuk tujuan praktis. Sebuah salinan harus disimpan dalam *ciphertext* sehingga asal dan isinya dapat diverifikasi seandainya ada bantahan. Cara yang lebih efisien untuk mencapai hasil yang sama adalah mengenkripsi blok kecil yang merupakan fungsi dokumen. Blok ini, dinamakan *authenticator*, harus memiliki sifat sulit untuk mengubah dokumen tanpa mengubah *authenticator*. Jika *authenticator* dienkripsi menggunakan kunci privat pengirim, ini akan berfungsi sebagai *signature* yang memverifikasi asal, isi , dan urutan.

Sangat penting untuk menekankan bahwa proses enkripsi yang baru saja dijelaskan tidak menyediakan keamanan. Pesan yang dikirim itu aman dari perubahan, tetapi tidak dari pencuri dengar. Hal ini jelas seandainya *signature* berdasarkan pada porsi pesan, karena sisa pesan disampaikan dengan jelas. Bahkan, walaupun enkripsi penuh pun, seperti yang ditujukkan dalam Gambar 3, tidak ada proteksi kerahasiaan karena orang yang melihat dapat mengenkripsi pesan dengan menggunakan kunci publik pengirim pesan.

Walaupun begitu, ada kemungkinan untuk menyediakan baik fungsi autentikasi maupun kerahasiaan dengan menggunakan skema kunci-publik ganda.

$$Z = E_{KU_b}[E_{KR_a}(X)]$$

$$X = D_{KU_a}[D_{KR_b}(Z)]$$

Dalam kasus ini, kita mulai seperti sebelumnya dengan mengenkripsi pesan, menggunakan kunci privat pengirim. Hal ini menyediakan *digital siganture*. Kemudian, kita mengenkripsi kembali menggunakan kunci publik penerima. *Ciphertext* akhir akan hanya dapat didekripsi oleh penerima yang dituju, yang memiliki kunci privat yang cocok. Jadi kerahasiaan tersedia. Kekurangan dari pendekatan ini adalah bahwa algoritma kunci publik, yang komplek, harus digunakan empat kali bukan dua dalam setiap komunikasi.

2.2. Aplikasi Kriptosistem Kunci-Publik

Sebelum melanjutkan, kita perlu mengkarifikasi salah satu aspek dari kriptosistem kunci-publik yang kalau tidak, akan menyebabkan kebingungan. Sistem kunci-publik dicirikan dengan penggunaan tipe kriptografik algoritma dengan dua kunci, satu yang dipegang pribadi dan satu yang lain dipublikasikan. Bergantung pada aplikasi, pengirim dapat menggunakan baik kunci privat pengirim maupun kunci publik penerima, atau keduanya, untuk melakukan

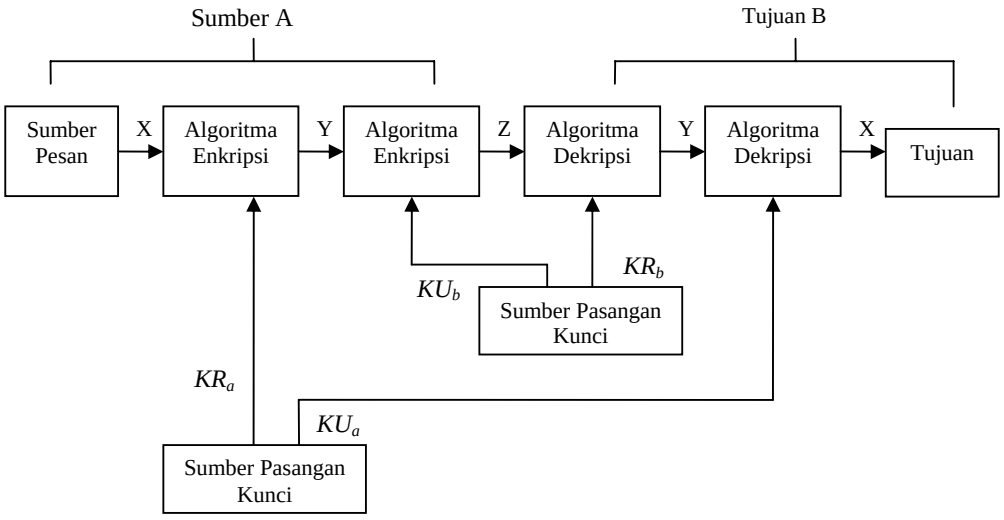
beberapa fungsi kriptografik. Dalam istilah luas, kita akan mengklasifikasi penggunaan kriptosistem kunci-publik dalam tiga kategori:

- ***Encryption/decryption*:** Pengirim mengenkripsi pesan menggunakan kunci publik penerima.
- ***Digital Signature*:** Pengirim "menandatangani" pesan dengan menggunakan kunci privatnya. Penandatanganan dicapai dengan algoritma kriptografik yang diterapkan pada pesan atau blok data yang merupakan fungsi pesan.
- ***Key exchange*:** Dua pihak yang bekerja sama bertukar kunci sesi. Beberapa pendekatan yang berbeda mungkin terjadi, melibatkan kunci privat salah satu pihak atau keduanya.

Beberapa algoritma sesuai untuk ketiga jenis aplikasi, walaupun beberapa yang lain hanya cocok untuk satu atau dua jenis aplikasi. Tabel 2 menunjukkan aplikasi yang didukung oleh algoritma yang didiskusikan dalam buku ini.

Algoritma	*Encrytion/Decryption*	*Digital Signature*	*Key Exchange*
RSA	Ya	Ya	Ya
Elliptic curve	Ya	Ya	Ya
Diffie-Hellman	Tidak	Tidak	Ya
DSS	Tidak	Ya	Tidak

Tabel 2 : Aplikasi Kriptosistem Kunci-Publik



Gambar 4: Kerahasiaan dan Autentikasi

2.3. Persyaratan Kriptografi Kunci-Publik

Kriptosistem yang telah diilustrasikan melalui Gambar 2 sampai Gambar 4 bergantung pada algoritma kriptografik yang berdasar pada dua kunci yang saling berhubungan. Diffie dan Hellman membuat postulat sistem ini tanpa mendemonstrasikan bahwa algoritma itu ada. Walaupun begitu, mereka meletakkan kondisi yang harus terpenuhi[DIFF76b]:

1. Secara komputasional, mudah untuk pihak B untuk membangkitkan pasangan kunci(kunci publik KU_b, kunci privat KR_b).
2. Secara komputasional, mudah untuk pengirim A, mengetahui kunci publik dan pesan yang akan dienkripsi, M, untuk membangkitkan *ciphertext* yang sesuai:

$$C = E_{KU_b}(M)$$

3. Secara komputasional, mudah bagi penerima B untuk mendekripsi *ciphertext* yang dihasilkan menggunakan kunci privat untuk mengembalikan pesan asli:

   $M = D_{KR_b}(C) = D_{KR_b}[E_{KU_b}(M)]$

4. Secara komputasional, sulit bagi musuh yang mengetahui kunci publik KU_b, untuk menentukan kunci privat, KR_b.
5. Secara komputasional, sulit bagi musuh yang mengetahui kunci publik KU_b dan *ciphertext*, C, untuk menentukan kunci privat, KR_b.

Kita dapat menambahkan persyaratan keenam yang walaupun berguna, tidak perlu untuk semua aplikasi kunci-publik.

6. Fungsi enkripsi dan dekripsi dapat diterapkan dalam urutan manapun:

   $M = E_{KU_b}[D_{KR_b}(M)]$

   $= D_{KU_b}[E_{KR_b}(M)]$

Ini adalah persyaratan berat, sebagaimana dibultikan oleh fakta bahwa hanya dua algoritma (RSA, elliptic curve) yang telah diterima luasdi beberapa dekade sejak konsep kriptografi kunci publik diusulkan.

Sebelum melanjutkan mengapa persyaratan tersebut berat, pertama mari kita membahas kembali persyaratan tersebut. Persyaratan menjadi kebutuhan untuk *trap-door one-way function*. *One-way function*(fungsi satu ke satu) adalah suatu fungsi yang memetakan domain ke range sedemikian sehingga setiap nilai fungsi memiliki invers yang unik, dengan kondisi bahwa kalkulasi fungsi mudah, sedangkan kalkulasi inverse adalah sulit.

$Y = f(X)$ mudah
$X = f^{-1}(Y)$ sulit

Secara umum, mudah didefinsikan bahwa masalah dapat diselesaikan dalam polinom waktu sebagai fungsi panjang input. Jadi, jika panjang masukan input adalah n bits, maka waktu untuk menghitung fungsi sebanding dengan n^a, di mana *a* adalah konstanta. Algoritma jenis ini dikatakan termasuk kelas **P**. Istilah sulit adalah konsep yang jauh lebih kompleks. Secara umum, kita dapat menyatakan suatu masalah sulit jika usaha untuk menyelesaikan masalah tersebut bertumbuh lebih cepat daripada fungsi polinomial. Misalnya, jika panjang input adalah n bits dan fungsi komputasi waktu sebanding dengan 2^n, masalah ini dianggap sulit. Walaupun begitu, sulit untuk menentukan bahwa algoritma tertentu menunjukkan kompleksitas ini. Lebih jauh lagi, ide kompleksitas komputasi tradisional menitikberatkan pada kasus terburuk atau kasus rata-rata kompleksitas suatu algoritma. Perhitungan semacam ini tak berguna dalam kriptografi, yang membutuhkan bahwa sulit untuk menginverskan sebuah fungsi untuk semua input, bukan untuk kasus terburuk ataupun kasus rata-rata.

Sekarang kita kembali pada definisi *trap-door one-way function*, yang mudah untuk dikalkulasi dalam satu arah dan sulit untuk dikalkulasi dalam arah sebaliknya kecuali kalau informasi tambahan diketahui. Dengan adanya informasi tambahan, inversnya dapat dikalkulasi dalam polinom waktu. Kita dapat merangkum sebagai berikut: *trap-door one-way function* adalah sekumpulan fungsi yang dapat diinverskan, sedemikian sehingga,

$Y = f_k(X)$ mudah, jika k dan X diketahui
$X = f_k^{-1}(Y)$ mudah, jika k dan Y diketahui
$X = f_k^{-1}(Y)$ sulit, jika hanya Y yang diketahui

Jadi, perkembangan skena kunci-publik yang praktis bergantung pada penemuan *trap-door one-way function* yang cocok.

2.4. Kriptanalisis Kunci-Publik

Sama seperti enkripsi simetris, skema enkripsi kunci-publik lemah terhadap serangan *brute-force*. Cara pemecahan masalahnya adalah sama: menggunakan kunci yang besar. Walaupun begitu, ada beberapa kompensasi yang harus diperhitungkan. Sistem kunci-publik begantung pada penggunaan beberapa jenis fungsi matematika yang dapat diinverskan. Kompleksitas dari penghitungan fungsi tersebut tidak bertambah secara linear terhadap jumlah bit, namun bertambah lebih cepat daripada itu. Jadi, ukuran kunci harus cukup besar untuk membuat serangan *brute-force* tidak praktis tetapi cukup kecil untuk membuat enkripsi dan dekripsi praktis. Secara singkat, ukuran kunci diusulkan untuk membuat setangan *brute-force*

tidak praktis tetapi hal ini berakibat pada kecepatan enkripsi dan dekripsi menjadi terlalu lambat untuk tujuan umum. Oleh karena itu, penggunaan enkripsi kunci-publik terbatas pada manajemen kunci dan aplikasi tandatangan (*signature application*).

Bentuk lain penyerangan adalah menemukan cara untuk menghitung kunci privat dari kunci publik yang diberikan. Sampai saat ini, belum terbukti secara matematika bahwa bentuk serangan ini adalah sulit untuk algoritma kunci-publik tertentu. Sejarah kriptanalisis menunjukkan bahwa sebuah masalah yang terlihat tidak dapat diselesaikan dari satu perspektif dapat ditemukan memiliki solusi dalam sudut pandang lain yang berbeda.

Terakhir, ada beberapa bentuk penyerangan yang khas terhadap sistem kunci-publik. Ini, pada dasarnya adalah serangan mencoba-coba membuka isi pesan. Anggap, misalnya, ada sebuah pesan yang dikirim yang berisi kunci 56 bit secara keseluruhan. Musuh dapat mengenkripsi semua pesan yang mungkin menggunakan kunci publik dan dapat mencoba-cpba mengartikan pesan dengan mencocokkan *ciphertext* yang dikirim. Jadi, tidak peduli berapa besat ukuran skema kunci-publik, serangan direduksi menjadi serangan *brute-force* pada kunci 56 bit. Serangan jenis ini dapat dicegah dengan menambahkan beberapa bilangan acak pada pesan sederhana.

3. *Diffie Hellman Key Exchange*

Algoritma pertama yang dipublikasikan muncul pada makalah oleh Diffie dan Hellman yang mendefinsikan kriptografi kunci publik [DIFF76b] dan secara umum disebut *Diffie-Hellman Key Exchange*. Sejumlah produk komersial menggunakan teknik ini.

Tujuan utama dari algoritma ini adalah membuat dua pengguna bertukar kunci secara aman sehingga kemudian dapat digunakan untuk enkripsi pesan. Algotima ini senfiri terbatas pada penukaran kunci.

Algoritma Diffie-Hellman bergantung pada efektifitas pada kesulitan menghitung logaritma diskret. Jelaasnya, kita mendefinisikan logaritma diskret dalam cara berikut. Pertama, kita mendefinisikan sebuah akar primitif (*primitive root*) dari sebuah bilangan prima *p* sebagai salah satu yang pangkatnya membengkitkan semua integer dari 1 sampai *p* - 1. Jika a adalah akar primitif dari sebuah bilangan prima *p*, maka bilangan-bilangan

$$a \bmod p,\ a^2 \bmod p, \ldots,\ a^{p-1} \bmod p$$

adalah berbeda dan terdiri dari bilangan mulai dari 1 sampai *p*-1.

Untuk integer sembarang *b* dan sebuah akar primitf *a* dari bilangan prima *p*, kita dapat menemukan eksponen unik i sedemikian sehinga

$$b \equiv a^i \bmod p \qquad \text{di mana } 0 \le i \le (p-1)$$

Pangkat *i* disebut sebagai logaritma diskret, atau index.dari *b* terhadap basis *a*, mod *p*. Nilai ini dinotasikan sebagai $\text{ind}_{a,p}(b)$.

Dengan latar ini, kita dapat mendefinisikan *Diffie-Hellman Key Exchange*, yang dirangkum dalam Gambar 5. Untuk skema ini, ada dua bilangan yang diketahui publik: sebuah bilangan prima *q* dan sebuah integer α yang merupakan akar primitif dari *q*. Anggap A dan B akan melakukan pertukaran kunci. A akan memilih bilangan integer acak $X_A < q$ dan menghitung $Y_A = \alpha^{X_A} \bmod q$. Sama halnya dengan B, B memilih bilangan integer acak $X_B < q$ dan menghitung $Y_B = \alpha^{X_B} \bmod q$. Masing – masing pihak menyimpan nilai privat *X* dan membuat nilai Y diketahui publik. Pengguna A akan menghitung kunci sebagai $K = (Y_B)^{X_A} \bmod q$ dan Pengguna B akan menghitung $K = (Y_A)^{X_B} \bmod q$. Masing-masing kalkulasi akan menghasilkan hasil yang identik.

$$\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha^{X_B})^{X_A} \bmod q \qquad \text{dari aturan modulus} \\
&= \alpha^{X_B X_A} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}$$

Hasilnya adalah kedua pihak telah bertukar kunci. Lebih jauh lagi, karena X_A dan X_B privat, musuh hanya memiliki bahan *q*, α, Y_A, dan Y_B. Jadi, musuh dipaksa untuk mengambil logaritma diskret untuk menentukan kunci.

Contohnya,menyerang kunci rahasia dari user B, musuh harus menghitung

$$X_B = \text{ind}_{\alpha,q}(Y_B)$$

Musuh kemudian dapat menghitung kunci K dengan cara yang sama seperti B.

Elemen Publik Global

q	bilangan prima
α	$\alpha < q$ dan α akar primitif dari q

Pembangkitan Kunci Pengguna A

Pilih privat X_A	$X_A < q$
Hitung Y_A	$Y_A = \alpha^{X_A} \bmod q$

Pembangkitan Kunci Pengguna B

Pilih privat X_B	$X_B < q$
Hitung Y_B	$Y_B = \alpha^{X_B} \bmod q$

Pembangkitan Kunci Rahasia Oleh A

$K = (Y_B)^{X_A} \bmod q$

Pembangkitan Kunci Rahasia Oleh B

$K = (Y_A)^{X_B} \bmod q$

Gambar 5 : Algoritma *Diffie Hellman Key Exchange*

Keamanan dari *Diffie-Hellman key exchange*(Penukaran kunci Diffie-Hellman) terletak pada fakta bahwa relatif mudah untuk menghitung eksponensial modulo sebuah bilangan prima tetapi sangat sulit menghitung logaritma diskret. Untuk bilangan prima yang lebih besar, bagian yang terakhir dianggap tidak mudah.

Berikut ini adalah contohnya. Penukaran kunci berdasarkan pada penggunaan bilangan prima q = 353 dan akar primitif dari 353, dalam hal ini α adalah 3. A dan B akan memilih kunci rahasia X_A = 97 dan X_B = 233. Masing-masing dihitung kunci publiknya:

$$Y_A = 3^{97} \bmod 353 = 40$$
$$Y_B = 3^{233} \bmod 353 = 248$$

Setelah mereka bertukar kunci publik, mereka menghitung kunci rahasia:

$A \rightarrow K = (Y_B)^{X_A} \bmod q = 248^{97} \bmod 353 = 160$

$B \rightarrow K = (Y_A)^{X_B} \bmod q = 40^{233} \bmod 353 = 160$

Kita asumsikan penyerang memiliki informasi berikut:

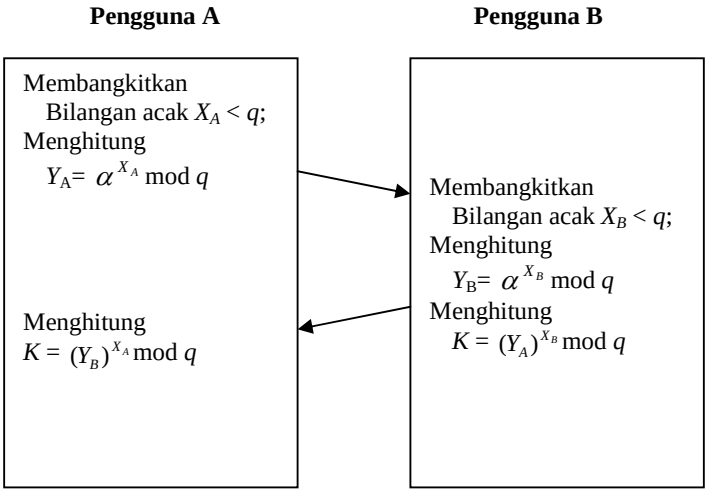
$$q = 353;\ \alpha = 3;\ Y_A = 40;\ Y_B = 248$$

Dalam contoh sederhana ini, masih mungkin untuk dilakukan *brute force* untuk menentukan kunci rahasia 160. Secara khusus, penyerang E dapat menentukan kunci umum dengan menemukan solusi persamaan $3^a \bmod 353 = 40$ atau persamaan $3^b \bmod 353 = 248$. Pendekatan *brute force* dalah menghitung pangkat dari 3 modulo 353, berhenti ketika hasil sama dengan 40 atau 248. Jawaban yang diinginkan dicapai ketika nilai eksponen 97, yang menghasilkan $3^{97} \bmod 353 = 40$.

Dengan angka yang lebih besar, masalah menjadi sulit dan tidak praktis.

Gambar 6 menunjukkan protokol sederhana yang menggunakan kalkulasi Diffie-Hellman. Anggap jika A menginginkan membangun koneksi dengan B dan menggunakan kunci rahasia untuk mengenkripsi pesan pada koneksi tersebut. A dapat membangkitkan kunci privat X_A, menghitung Y_A, dan mengirimkannya kepada B. B merespon dengan membangkitkan nilai privat X_B, menghitung Y_B, dan mengirim Y_B kepada A. Masing-masing pengguna sekarang dapat menghitung kunci. Nilai publik q dan α perlu diketahui sebelumnya. Alternatif lainnya, A dapat mengambil nilai untuk q dan α dan memasukkannya dalam pesan pertama.

Contoh dari penggunaan algoritma Diffie-Hellman lainnya, Anggap sekelompok *user* (misal: semua pengguna LAN) masing-masing membangkitkan nilai privat X_A dan menghitung Y_A. Nilai publik ini, bersama dengan nilai publik global untuk q dan α, disimpan pada beberapa direktori sentral. Pada waktu kapan pun, B dapat mengakses nilai publik(*public value*) A, menghitung kunci rahasia, dan menggunakannya untuk mengirim pesan yang telah dienkripsi kepada A. Jika direktori pusat terpercaya, maka bentuk komunikasi ini menyediakan baik kerahasiaan maupun autentikasi. Karena hanya A dan B yang menentukan kunci, tidak ada pengguna lain yang dapat mebaca pesan tersebut (kerahasiaan). Penerima A mengetahui bahwa hanya B yang dapat membuat pesan tersebut menggunakan kunci ini (autentikasi).



Gambar 6: ***Diffie Hellman Key Exchange***

4. Kesimpulan

Kesimpulan yang dapat diperoleh dari Aplikasi Matematika Diskret Teori Bilangan Dalam Kriptografi Kunci Publik dan Algoritma Diffie-Hellman adalah:

1. Tidak ada kriptografi yang benar-benar aman untuk setiap serangan atau dari kriptanalisis.
2. Ada perbedaan mencolok dari kriptografi simetris dan Kriptografi Kunci Publik
3. Digital Signature hanya menyediakan autentikasi, tidak menyediakan kerahasiaan isi. Namun, kita bisa meyediakan kerahasiaan dengan cara mengenkripsi pesan tersebut dengan kunci publik orang yang dituju.
4. Algoritma Diffie-Hellman menggunakan Teori Bilangan dalam pembuatan kunci rahasia. Namun, algoritma ini terbatas pada pertukaran kunci saja.
5. Algoritma Diffie-Hellman meyediakan baik autentikasi maupun kerahasiaan jika direktori sentral terpercaya.

DAFTAR PUSTAKA

[1] Diffie, W., and Hellman, M. (1976). ”Multiuser Cryptographics Techniques”. *IEEE Transactions on Information Theory*.

[2] Diffie-Hellman key exchange - Wikipedia, the free encyclopedia. (2006). http://en.wikipedia.org/wiki/Diffie-Hellman. Tanggal Akses: 26 Desember 2006 pukul 15:10.

[3] Public-key cryptography - Wikipedia, the free encyclopedia. (2006). http://en.wikipedia.org/wiki/Public-key_cryptography. Tanggal Akses: 26 Desember 2006 pukul 15:00.

[4] Cryptography FAQ (06/10: Public Key Cryptography). (2006). *http://www.faqs.org/faqs/cryptography-faq/part06/*. Tanggal.akses: 26 Desember 2006 pukul 15:20.

[5] Stallings, William. (2003). Cryptography and Network Security. New Jersey: Pearson Education, Inc.