SISTEM PENYANDIAN RSA

SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Pendidikan
Program Studi Pendidikan Matematika

Disusun Oleh :

Rosa Maria

NIM : 88 414 019
NIRM : 880052010501120018

JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS PENDIDIKAN MATEMATIKA DAN ILMU PENGETAHUAN ALAM
IKIP SANATA DHARMA
YOGYAKARTA
1 9 9 3

S k r i p s i

Sistem Penyandian RSA

Oleh

Rosa Maria

NIM : 88 414 019

NIRM : 8800520105011200l8

telah disetujui oleh:

Pembimbing I

Dr. F.Susilo, SJ tanggal 31-3-1993

Pembimbing II

Dra. Linda Yuliastuti tanggal 31-3-1993

S K R I P S I

SISTEM PENYANDIAN RSA

yang dipersiapkan dan disusun oleh

Rosa Maria

NIM : 88 414 019

NIRM : 880052010501120018

telah dipertahankan di depan Panitia Penguji

pada tanggal 31 Maret 1993

dan dinyatakan telah memenuhi syarat

Susunan Panitia Penguji

	Nama lengkap	Tanda tangan
Ketua	Dr. St. Suwarsono	
Sekretaris	Drs. F. Kartika Budi, M Pd	
Anggota	Dr. F. Susilo, SJ	
Anggota	Dra. Y. Linda Yuliastuti	
Anggota	Drs. A. Tutoyo, Msc	
Anggota	Prof. Drs. Wirasto	

Yogyakarta, 31 Maret 1993

Fakultas PMIPA

IKIP Sanata Dharma

Dekan

Dr. St. Suwarsono

kupersembahkan untuk
kebahagiaan papa, mama dan ketiga kakakku

KATA PENGANTAR

Puji dan syukur penulis haturkan pada Tuhan, karena atas kasih dan rahmatNya, penulis dapat menyelesaikan skripsi ini. Adapun penyusunan skripsi ini dimaksudkan untuk memenuhi salah satu syarat memperoleh gelar sarjana pendidikan di IKIP Sanata Dharma Yogyakarta. Judul skripsi ini adalah "SISTEM PENYANDIAN RSA".

Penulis menyadari bahwa tulisan ini masih jauh dari sempurna dan tidak lepas dari kesalahan dan kekurangan, mengingat kemampuan dan pengetahuan penulis masih sangat terbatas.

Penyusunan skripsi ini tentu tidak akan terwujud tanpa petunjuk, bimbingan dan bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. St. Suwarsono, selaku Ketua Jurusan Pendidikan Matematika IKIP Sanata Dharma Yogyakarta yang telah merestui penulisan skripsi ini.
2. Romo Dr. F. Susilo, SJ, selaku Dosen Pembimbing I yang di tengah-tengah kesibukannya selalu meluangkan

waktu untuk membimbing dan memberikan petunjuk hingga selesainya skripsi ini.

3. Ibu Dra. Linda Yuliastuti, selaku Dosen Pembimbing II yang telah meluangkan waktu untuk memberikan bimbingan kepada penulis.
4. Papa, mama dan kakak-kakak yang selalu memberikan dorongan moril dan memenuhi setiap biaya yang diperlukan dalam menunjang penyusunan skripsi ini.
5. Para sahabat yang telah mendukung dan selalu memberikan semangat serta saran-saran yang dibutuhkan penulis.

Akhirnya penulis berharap agar penulisan skripsi ini dapat memberikan motivasi bagi rekan-rekan yang masih menimba ilmu dan dapat bermanfaat bagi kita semua.

Penulis

DAFTAR ISI

ABSTRAK

SISTEM PENYANDIAN RSA

Dalam tulisan ini disajikan sistem penyandian yang dirancang pada tahun 1977 oleh Rivest, Shamir dan Adleman, dan dikenal dengan nama "Sistem Penyandian RSA".

Dalam sistem penyandian ini, pengirim dan penerima pesan memilih dua buah bilangan prima p dan q yang berbeda dan dirahasiakan, serta sebuah bilangan bulat a yang prima relatif terhadap (p-1)(q-1), di mana $0 < a < (p-1)(q-1)$. Bilangan a dan $n = pq$ tidak perlu dirahasiakan.

Dengan menggunakan suatu tabel yang telah disepakati, pesan yang akan dikirimkan diubah ke dalam bentuk desimal menjadi barisan bilangan bulat M_i di mana $0 < M_i < n$.

Pesan yang dikirimkan adalah barisan bilangan bulat R_i, yang didapat dari persamaan $R_i \equiv M_i^a \pmod{n}$.

Untuk membaca pesan sandi tersebut si penerima pesan harus menghitung invers dari a, namakan d, dalam modulo (p-1)(q-1), yaitu $ad \equiv 1 \pmod{(p-1)(q-1)}$. Masing-masing M_i diperoleh dari persamaan $M_i \equiv R_i^d \pmod{n}$. Barisan bilangan bulat M_i dibaca dengan menggunakan tabel yang sama.

BAB I

P E N D A H U L U A N

A. Latar Belakang Masalah

Aturan komunikasi rahasia yang dipakai oleh dua individu dinamakan sistem penyandian. Sistem penyandian bukanlah merupakan suatu sistem yang baru. Sejak zaman dulu orang-orang Indian kuno telah mengenal suatu sistem penyandian, walaupun hanya merupakan sistem penyandian yang masih sederhana, seperti pengkodean dengan menggunakan api, asap atau bunyi-bunyian tertentu. Biasanya digunakan satu atau beberapa kunci untuk menjaga kerahasiaan sebuah komunikasi. Kunci ini merupakan barisan berhingga huruf atau angka yang telah disepakati oleh pemakai sistem penyandian untuk membuat serta membaca pesan sandi.

Seiring dengan perkembangan zaman, orang terus berusaha untuk mengembangkan sistem penyandian yang lebih baik dan yang dapat menjamin kerahasiaan suatu komunikasi. Mulai dari sistem komunikasi sederhana yang biasa digunakan untuk kasus-kasus yang sederhana pula, misal-

nya yang biasa digunakan oleh para pramuka, sampai pada sistem penyandian yang lebih rumit, dan yang lebih dapat menjamin kerahasiaan suatu komunikasi, dan tentu saja dapat dimanfaatkan untuk kasus-kasus yang lebih rumit pula, misalnya dalam dunia militer dan diplomatik.

Salah satu sistem penyandian yang baru ditemukan pada tahun 1977 oleh Rivest, Shamir dan Adleman atau lebih dikenal dengan nama sistem penyandian RSA, merupakan suatu sistem penyandian yang dapat diandalkan, karena dalam sistem penyandian ini kita dapat mengandaikan bahwa pesan sandi yang kita kirimkan pada pihak tertentu dapat disadap oleh ahli cryptanalysis, yaitu ahli yang berusaha untuk membaca pesan sandi, walaupun tidak mengetahui parameter kunci dari sistem penyandiannya. Juga diandaikan bahwa pihak lawan telah menguasai sistem penyandian RSA ini. Sedangkan yang menjadi parameter kunci adalah dua buah bilangan yang harus memenuhi syarat - syarat tertentu.

Sistem penyandian RSA ini menjadi lebih menarik lagi, karena sistem ini menggunakan terapan dari teori bilangan, seperti Algoritma Euclides, untuk menentukan faktor persekutuan terbesar dari dua bilangan bulat, bilangan modulo dan bilangan prima.

B. Perumusan Masalah

Pokok permasalahan yang akan diteliti adalah : Bagaimana sistem penyandian RSA itu ?

Untuk sampai pada pokok permasalahan tersebut akan dibahas dulu beberapa sub masalah berikut :

1. Apa yang menjadi kriteria sebuah sistem penyandian itu dapat dikatakan aman bagi sipengirim dan sipenerima pesan ?
2. Theorema-theorema mana dari Teori Bilangan yang mendasari sistem penyandian RSA ?
3. Bagaimana bukti dari theorema - theorema tersebut ?

Setelah menjawab seluruh permasalahan di atas, kiranya perlu dibahas lagi sub masalah berikut :

Kesulitan apa yang dihadapi ahli cryptanalysis untuk memecahkan sistem penyandian RSA ?

C. Tujuan Penulisan

Tujuan dari penulisan ini adalah untuk menjelaskan dan membahas sistem penyandian RSA yang akan dimulai dengan menjelaskan tentang kriteria sistem penyandian

yang aman, serta landasan teori yang mendukung sistem penyandian tersebut .

D. Pembatasan Masalah

Perlu ditegaskan bahwa tulisan ini hanya akan menyajikan dengan lengkap dan jelas tentang sistem penyandian RSA serta teori-teori yang menunjangnya, sedangkan beberapa sistem penyandian lain hanya akan disajikan sebagai contoh guna dibandingkan dengan sistem penyandian RSA.

E. Manfaat Penulisan

Manfaat umum dari penulisan ini adalah sebagai contoh penerapan teori bilangan, yang sebagian landasan teorinya telah dipelajari di bangku kuliah .

Sedangkan manfaat khususnya adalah dapat digunakannya sistem penyandian RSA ini jika seorang sarjana matematika terjun ke bidang-bidang yang memerlukan penyandian, misalnya dunia militer dan diplomatik. Di samping itu tulisan ini diharapkan juga dapat memotivasi

para matematikawan untuk terus mengembangkan penerapan matematika, walaupun matematika itu merupakan ilmu yang abstrak .

BAB II

CRYPTOLOGY

A. Pengantar pada Sistem Penyandian

Cryptology adalah suatu ilmu yang mempelajari tentang sistem - sistem komunikasi rahasia. Ada dua bidang keilmuan yang terdapat dalam Cryptology yaitu cryptography dan cryptanalysis, yang saling melengkapi satu sama lain. Cryptography merupakan suatu studi untuk merancang sistem penyandian yang aman dan cryptanalysis merupakan suatu studi untuk memecahkan suatu sistem penyandian.

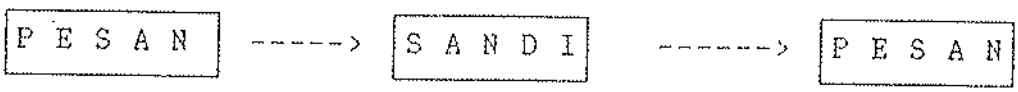
Jika kita ingin mengirim suatu pesan dengan sandi, kita harus yakin, bahwa pesan - pesan kita itu akan aman, walaupun pesan itu sampai ke tangan seorang ahli cryptanalysis. Jadi dengan begitu seorang ahli cryptography juga harus mengetahui tentang cryptanalysis, karena cara terbaik untuk mengetahui bahwa suatu sistem penyandian itu aman, adalah dengan mencoba memecahkan sendiri sistem penyandian tersebut, serta mencoba meng-

hitung dengan berapa cara suatu sistem penyandian itu dapat dipecahkan. Semakin banyak cara atau semakin tak berhingga cara yang harus dicoba oleh ahli cryptanalysis untuk memecahkan suatu sistem penyandian tentu semakin baiklah sistem penyandian tersebut.

B. Sistem Penyandian

Tentu saja ada alasan yang cukup kuat bagi seseorang untuk memutuskan bahwa suatu pesan harus disandikan dulu sebelum dikirimkan. Dan dengan berdasarkan alasan tersebutlah orang dapat menentukan sistem penyandian mana yang aman baginya. Seandainya alasan orang untuk menyandikan suatu pesan cukup sederhana saja maka tentu ia akan memilih sistem penyandian yang sederhana pula, sebaliknya jika alasan orang untuk menyandikan suatu pesan cukup kuat, maka tentu orang akan memilih sistem penyandian yang lebih rumit dan yang benar-benar dapat menjamin kerahasiaan pesan tersebut.

Bagan dari sebuah sistem penyandian yang biasa digunakan adalah :



Pengirim mengirimkan pesan kepada penerima dengan terlebih dahulu mengubah pesan tersebut menjadi pesan sandi dan kemudian sipenerima akan menerjemahkan kembali pesan sandi tersebut menjadi pesan biasa. Pesan sandi ini dapat bermacam-macam bentuknya, ada yang berupa huruf-huruf yang telah dikacaukan, ada yang berupa tanda - tanda atau gambar-gambar, sedangkan yang akan kita bahas dalam tulisan ini adalah pesan sandi yang berupa angka-angka.

Tentu saja ada hal-hal yang harus diperhatikan oleh pengirim dan penerima pesan yaitu bahwa harus ada kesesuaian dalam memilih sebuah sistem penyandian dan harus menggunakan parameter kunci yang sesuai.

Khusus untuk pesan-pesan yang bersifat sangat rahasia perlu diperhatikan hal-hal berikut, yaitu kita harus mengandaikan bahwa pesan-pesan itu dikirimkan pada jalur komunikasi yang tidak aman (dapat disadap), dan para ahli cryptanalysis telah mengenal setiap sistem penyandian. Jadi para ahli cryptanalysis hanya tinggal menemukan kembali parameter kunci yang digunakan.

Biasanya semakin banyak parameter kunci yang digunakan semakin amanlah sebuah sistem penyandian, dan tentu saja semakin kurang menyenangkan untuk digunakan. Kasus semacam ini sama dengan kunci kombinasi pada sebuah

lemari besi. Sebuah lemari besi akan lebih aman jika menggunakan banyak kombinasi angka-angka, tapi tentu semakin sulit bagi sipemakai untuk menghafalkan angka-angka tersebut.

C. Beberapa Contoh Sistem Penyandian yang Sederhana

Di antara metode - metode yang termudah dan sekaligus merupakan metode yang tertua adalah Sandi Caesar. Dalam metode ini setiap huruf dan angka ke-n dari tabel diganti dengan huruf dan angka ke-(n+k) dari tabel, di mana k adalah suatu bilangan bulat tertentu yang telah disepakati oleh orang-orang yang akan mengirim dan menerima sandi. Misalnya jika diambil k = 10, akan diperoleh tabel berikut:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		1	2	3	4	5	6	7
Y	Z		1	2	3	4	5	6	7	8	9												
8	9	A	B	C	D	E	F	G	H	I	J												

tabel 2.1 Sandi Caesar dengan k=10

Jadi

P e s a n : IKIP SANATA DHARMA YOGYAKARTA
Teks sandi : SUSZA2KXK3KANRK1WKA8YQ8KUK13K

Sandi Caesar ini dapat dikembangkan sendiri oleh sipemakai, misalnya dengan menambahkan atau menggantikan

angka-angka tersebut dengan lambang - lambang yang diperlukan dalam menyandikan pesannya. Metode ini lemah, karena ahli cryptanalysis hanya harus mengira atau menduga nilai k, dengan mencoba masing-masing dari 36 pilihan untuk dapat membaca pesan tersebut.

Ada sebuah metode sederhana lain yang lebih baik, yaitu dengan menggunakan suatu tabel untuk menentukan penggantian setiap huruf menjadi sandi. Contoh diberikan tabel korespondensi 1 - 1 :

ABCDEFGHIJKLMNOPQRSTUVWXYZ
THE QUICKBROWNFXJMPDVRLAZVG

maka pesan tersebut di atas dapat disandikan sebagai berikut:

P e s a n : IKIP SANATA DHARMA YOGYAKARTA
Teks sandi : BOBJTDHFHVHTQKHPNHTVXCVHOHPVH

Metode ini lebih baik jika dibandingkan dengan metode yang pertama tadi, karena ahli cryptanalysis harus mencoba lebih banyak untuk memecahkan pesan tersebut, yaitu kira-kira 27! (yang lebih besar dari 10^{28}) tabel. Walaupun begitu metode ini masih mempunyai kekurangan yaitu adanya suatu sifat yang melekat dalam suatu bahasa. Misalnya dalam bahasa Inggris huruf E mempunyai freku-

ensi muncul yang tinggi, ditambah lagi tidak adanya kombinasi QJ, sedangkan kombinasi ER sangat sering dipergunakan. Dengan mempelajari frekuensi munculnya huruf-huruf dan kombinasinya, pemecahan teks sandi tersebut akan dipermudah.

Ada lagi suatu metode agar para ahli cryptanalysis lebih sulit untuk memecahkan teks sandi tersebut. Metode ini merupakan perluasan dari metode Sandi Caesar dan dikenal dengan nama metode Sandi Vigenere. Dalam Sandi Vigenere digunakan sebuah kunci secara berulang-ulang. Untuk menemukan teks sandinya, digunakan indeks suatu huruf yaitu urutan huruf tersebut dalam abjad. Misalnya huruf A dikatakan berindeks 1, huruf B berindeks 2 dan seterusnya sampai huruf Z yang berindeks 26 dan spasi dianggap berindeks 0. Cara menemukan teks sandinya adalah dengan menambahkan indeks huruf kunci dengan indeks huruf dalam pesan yang bersesuaian. Misal digunakan kunci ABC, maka penyandian pesan ini menjadi :

K u n c i : ABCABCABCABCABABCABCABCABCABC
P e s a n : IKIP SANATA DHARMA YOGYAKARTA
Teks sandi : JMLQBWBPDUCCEJBTPBBBPIBBMDSVD

Misalnya huruf terakhir dari teks pesan tersebut adalah A, dengan indeks 1, huruf kunci yang bersesuaian yaitu C, dengan indeks 3, maka huruf pesan teks sandi yang

bersesuaian ialah D (yang indeksnya 4).

Pada umumnya contoh-contoh di atas masih mempunyai banyak keterbatasan, umpamanya untuk pesan yang memuat lambang bilangan atau tanda baca. Seperti dalam sandi Caesar dengan kesepakatan, kita dapat saja mencantumkan lambang-lambang tertentu dalam tabel. Tapi pada sistem penyandian lainnya (dalam contoh di atas) kita akan mengalami kesulitan ataupun ketidakpraktisan. Umpamanya jika akan mengirimkan pesan: "pukul 8.10", maka kita harus menuliskannya dengan "pukul delapan lewat sepuluh menit".

BAB III

LANDASAN TEORI YANG MENUNJANG SISTEM PENYANDIAN RSA

Sejak seseorang mulai belajar matematika, padanya mulai dikenalkan operasi-operasi aritmatika seperti penjumlahan, pengurangan, perkalian, pembagian, atau operasi lainnya yang lebih kompleks. Dalam bab ini akan diperlihatkan bagaimana operasi-operasi aritmatika yang telah sangat dikenal itu dengan ditunjang oleh definisi, lemma, theorema, serta corollary dari teori bilangan dapat menjamin suatu sistem penyandian, hingga sungguh-sungguh aman bagi si pemakainya.

Semesta pembicaraan dalam bab ini adalah himpunan semua bilangan bulat yang biasa dilambangkan dengan Z. Hanya kadang-kadang saja kita menekankan pada bilangan bulat positif.

A. Faktor Persekutuan Terbesar

Sebelum kita membicarakan atau mendefinisikan faktor persekutuan terbesar (fpb), akan didefinisikan

dulu faktor dari suatu bilangan bulat, serta faktor persekutuan dari dua bilangan bulat.

3.1.1. Definisi Faktor :

Jika $a,b \in Z$ dengan $b \neq 0$, dikatakan b merupakan faktor dari a jika ada $r \in Z$ sedemikian sehingga $a = br$.

Biasanya ditulis "$b|a$" untuk menyatakan bahwa b merupakan faktor dari a.

3.1.2. Definisi Faktor Persekutuan :

Untuk sebarang $a,b \in Z$, $h \in Z$ dikatakan faktor persekutuan dari a dan b jika $h|a$ dan $h|b$.

3.1.3. Definisi Faktor Persekutuan Terbesar (fpb) :

Jika $b,c \in Z$ dan $b,c \neq 0$, dan d merupakan bilangan bulat positif terbesar sedemikian sehingga $d|b$ dan $d|c$, maka d disebut faktor persekutuan terbesar dari b dan c, dan ditulis dengan lambang $d = fpb(b,c)$.

Dalam penyajian selanjutnya jika dikatakan faktor persekutuan terbesar maka yang dimaksudkan adalah faktor persekutuan terbesar dari dua buah bilangan bulat positif, walaupun sebenarnya kita dapat saja menentukan faktor persekutan terbesar dari n buah bilangan bulat. Alasannya hanya membicarakan faktor persekutuan terbesar dari dua buah bilangan bulat positif adalah karena teori-teori yang akan disajikan dalam bab ini akan diarahkan pada suatu terapan, yaitu sistem penyandian RSA yang hanya menggunakan faktor persekutan terbesar dari dua buah bilangan bulat positif.

Apakah setiap pasang bilangan bulat positif itu mempunyai faktor persekutuan terbesar ? Setiap pasang bilangan bulat positif pasti mempunyai 1 sebagai faktor persekutuannya dan bilangan bulat terbesar yang mungkin menjadi faktor persekutuan dari b dan c adalah Min{b,c}, yaitu bilangan yang paling kecil dari antara b dan c itu sendiri, karena fpb(b,c) haruslah faktor dari b dan c, dan faktor dari b dan c tidak mungkin lebih besar dari b atau c itu sendiri. Jadi setiap pasang bilangan bulat

positif b dan c pasti mempunyai faktor persekutuan terbesar yang nilainya berkisar mulai dari 1 sampai dengan bilangan yang paling kecil dari antara b dan c itu sendiri.

Pada bagian berikut ini akan diuraikan tentang "Algoritma Euclides" untuk menentukan faktor persekutuan terbesar dari dua buah bilangan bulat.

B. Algoritma Euclides

Misalkan kita akan menentukan faktor persekutuan terbesar dari dua buah bilangan bulat positif b dan c, di mana $b \leq c$. Jika d merupakan faktor dari b dan c, (jadi mungkin d = fpb(b,c)), maka d merupakan faktor dari c-b. Jika d merupakan faktor dari c-b dan b, maka d juga merupakan faktor dari c. Tentu saja akan lebih mudah bagi kita untuk menghitung fpb(b,c-b) dari pada menghitung fpb(b,c), karena c-b lebih kecil.

Jika c-b lebih kecil dari c, maka c-2b masih lebih kecil lagi dari c, dan demikian pula seterusnya untuk c-3b,c-4b,..., c-qb. Jadi akan lebih baik jika kita dapat mengurangi terus c dengan qb yang mungkin, sehingga menghasilkan bilangan yang tidak negatif.

Sekarang permasalahannya adalah bagaimana kita dapat menentukan bilangan bulat q yang terbesar sedemikian sehingga $c - qb \geq 0$. Permasalahan ini sebenarnya merupakan dasar dari Algoritma Euclides, untuk menentukan fpb (b,c).

Jika kita membagi c dengan b, maka akan didapat hasil bagi q_1 dan sisanya r_1 :

$$\frac{c}{b} = q_1 + \frac{r_1}{b}$$

atau dapat juga ditulis:

$$c = q_1 b + r_1 \quad \text{.....(A)}$$

di mana r_1 harus memenuhi $0 \leq r_1 < b$.

3.2.1. Lemma:

Jika $b,c,q,r \in Z$ sedemikian sehingga $c = qb + r$, maka $fpb(b,c) = fpb(b,r)$.

Bukti:

Karena suatu bilangan bulat yang membagi b dan c juga membagi baik b maupun r, maka fpb(b,c) membagi b dan r, dan dengan demikian paling besar sama dengan fpb(b,r).

Maka $fpb(b,c) \leq fpb(b,r)$ .....(i)

Suatu bilangan bulat yang membagi b dan r juga

membagi c.

Jadi fpb(b,r) ≤ fpb(b,c) .....(ii)

Dari persamaan (i) dan (ii) didapat:

fpb(b,c) = fpb(b,r).

Jika dalam persamaan (A) : $r_1 = 0$ maka $c = q_1 b$ dan fpb(b,c) = b. Tapi jika $r_1 > 0$ maka kita masih harus menghitung faktor persekutuan terbesar dari b dan r_1. Tapi permasalahannya menjadi lebih sederhana, karena kita bekerja dengan bilangan yang lebih kecil. Selanjutnya cara tersebut diulangi lagi, yaitu kita membagi b dengan r_1, sehingga diperoleh hasil bagi yang baru yaitu q_2 dan sisa yang baru yaitu r_2 :

$$b = q_2 r_1 + r_2 \quad \text{dengan} \quad 0 \leq r_2 < r_1.$$

Jika $r_2 = 0$, maka r_1 faktor dari b dan

$r_1 = fpb(r_1,b) = fpb(b,c)$ ..... menggunakan lemma 3.2.1.

Secara umum (juga bila $r_2 \neq 0$), dengan menggunakan lemma 3.2.1, akan didapat :

$$fpb(b,c) = fpb(r_1,b) = fpb(r_2,r_1).$$

Selanjutnya kita membagi r_1 dengan r_2, lalu r_2 dengan r_3, dan seterusnya sampai kita memperoleh sisa nol. Semua pengerjaan di atas dapat dirangkum dalam persamaan -

persamaan di bawah ini, yang akan disebut persamaan - persamaan Euclides.

3.2.2. Persamaan-Persamaaan Euclides:

$$c = q_1 b + r_1 \quad \text{dengan } 0 \le r_1 < b$$

$$b = q_2 r_1 + r_2 \quad \text{dengan } 0 \le r_2 < r_1$$

$$r_1 = q_3 r_2 + r_3 \quad \text{dengan } 0 \le r_3 < r_2$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$r_{i-2} = q_i r_{i-1} + r_i \quad \text{dengan } 0 \le r_i < r_{i-1}$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$r_{k-3} = q_{k-1} r_{k-2} + r_{k-1} \quad \text{dengan } 0 \le r_{k-1} < r_{k-2}$$

$$r_{k-2} = q_k r_{k-1} + 0 \quad \text{dengan } r_k = 0$$

Menurut Euclides, fpb(b,c) adalah sisa terakhir yang tidak nol dalam barisan persamaan-persamaan Euclides di atas, yaitu r_{k-1}. Hal ini nanti akan dibuktikan pada theorema 3.2.3. Sebelum membuktikan theorema tersebut, perlu diperhatikan bagaimana persamaan-persamaan Euclides tersebut akan berhenti. Semua sisa dari persamaan-persamaan Euclides tersebut memenuhi :

$$b > r_1 > r_2 > r_3 > \ldots > r_{i-1} > r_i > \ldots > r_{k-1}$$

dan semua sisa itu harus merupakan bilangan bulat yang tidak negatif. Jadi pada akhirnya sisanya haruslah nol.

3.2.3 Theorema:

Bila b dan c adalah bilangan-bilangan bulat positif, maka dengan Persamaan-Persamaan Euclides, sisa terakhir yang tidak nol sama dengan fpb(b,c).

Bukti:

Andaikan $b,c \in Z^+$, maka harus dibuktikan bahwa:

$$fpb(b,c) = r_{k-1}.$$

(lihat persaman-persamaan Euclides di atas).

Kita perhatikan persamaan yang terakhir dari persamaan-persamaan Euclides $r_{k-2} = q_k r_{k-1} + 0$. Tampak bahwa r_{k-1} merupakan faktor dari r_{k-2}, karena $r_{k-2}/r_{k-1} = q_k$ (di mana q_k adalah bilangan bulat). Jadi :

$$r_{k-1} = fpb(r_{k-2}, r_{k-1}).$$

Dengan menggunakan lemma 3.2.1 dan dengan memakai dua persamaan terakhir dari persamaan - persamaan Euclides didapat :

$$fpb(r_{k-3}, r_{k-2}) = fpb(r_{k-2}, r_{k-1}) = r_{k-1}.$$

Lanjutkan dan ulangi terus pemakaian Lemma 3.2.1, maka akan didapat :

$$\begin{aligned} fpb(b,c) &= fpb(b,r_1) \\ &= fpb(r_1,r_2) \\ &= fpb(r_2,r_3) \\ &\ldots \end{aligned}$$

$= fpb(r_{k-2}, r_{k-1})$

$= r_{k-1}$. ∎

Contoh 3.1: Jika kita akan menentukan fpb(26,32), maka dengan menggunakan theorema 3.2.3 diperoleh:

$$32 = 1.26 + 6$$

$$26 = 4.6 + 2$$

$$6 = 3.2 + 0$$

Jadi fpb(26,32) = 2, di mana 2 merupakan sisa terakhir yang tidak nol.

3.2.4. Corollary

Jika $g = fpb(b,c)$ maka ada bilangan - bilangan bulat x dan y sedemikian sehingga $g = xb + yc$.

Bukti :

Perhatikan persamaan - persamaan Euclides tersebut di atas. Dari persamaan yang pertama diperoleh $r_1 = c - q_1 b$. Jika $g = r_1$, maka corollary ini telah terbukti.

Jika $g \neq r_1$, maka dengan menggunakan persamaan Euclides yang kedua, didapat :

$r_2 = b - q_2 r_1$

$= b - q_2 (c - q_1 b)$ subsitusi

$= b - q_2 c + q_1 q_2 b$ distribusi

$= (1+q_1q_2)b - q_2c$ distribusi

Proses ini dilanjutkan terus sampai :

$$g = r_{k-1} = r_{k-3} - q_{k-1}r_{k-2}$$

dan untuk r_{k-3} dan r_{k-2} disubsitusikan bentuk-bentuk yang telah diperoleh sebelumnya, dan kemudian nyatakan $g = r_{k-1}$ ke dalam bentuk xb+yc.

Persamaan tersebut di atas menyatakan fpb(b,c) ini sebagai kombinasi linier dari b dan c. Bukti dalam corollary ini nanti akan dipakai dalam sistem penyandian RSA yang akan dibahas pada Bab IV.

Contoh 3.2: Telah diperoleh bahwa fpb(26,32) = 2. Selanjutnya akan digunakan bukti dalam corollary 3.2.4. untuk menyatakan 2 sebagai kombinasi linier dari 26 dan 32.

Dari persamaan Euclides yang pertama diperoleh :

$$6 = 1.32 - 1.26 \quad \ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Dari persamaan Euclides yang kedua diperoleh :

$$2 = 1.26 - 4.6 \quad \ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Subsitusikan persamaan (1) ke dalam persamaan (2) :

$$2 = 1.26 - 4\ (1.32 - 1.26)$$

$$2 = 5.26 - 4.32 \ .$$

C. Kongruensi dan Relasi Ekuivalensi

Aritmetika bilangan bulat merupakan dasar dari teori bilangan. Salah satu bagian khusus dari teori bilangan adalah aritmetika modulo yang didasari atas kongruensi, yang nanti akan dipakai dalam sistem penyandian RSA.

3.3.1. Definisi :

Jika $n \in Z^+$ dan $a,b \in Z$, maka kita definisikan $a \equiv b \pmod{n}$ (baca : a kongruen b modulo n), jika ada $k \in Z$ sedemikian sehingga $a-b = kn$.

Dalam kehidupan sehari-hari kita sering menggunakan bilangan bulat modulo 12. Misalnya kalau kita memulai suatu pekerjaan pada pukul 10 pagi dan baru akan selesai dalam waktu 4 jam, maka kita akan mengatakan bahwa kita selesai pada pukul 2 siang, atau $10+4 = 14 \equiv 2 \pmod{12}$.

3.3.2. Definisi Relasi Ekuivalensi :

Suatu relasi $\sim$ yang didefinisikan pada himpunan S disebut relasi ekuivalensi jika untuk tiap a,b dan $c \in S$ berlaku :

(i) $a \sim a$ (sifat refleksif)

(ii) jika a ~ b, maka b ~ a (sifat simetris)

(iii) jika a ~ b dan b ~ c, maka a ~ c (sifat transitif).

3.3.3. Lemma:

Relasi "kongruensi modulo n" yang didefinisikan pada Z adalah suatu relasi ekuivalensi.

Bukti:

(i) ambil sebarang a ∈ Z, maka ada 0 ∈ Z sedemikian sehingga a - a = 0n, jadi kita peroleh a ≡ a (mod n) (sifat refleksif).

(ii) ambil sebarang a,b ∈ Z.

Jika a ≡ b (mod n), maka ada bilangan bulat k, sedemikian sehingga a - b = kn. Jadi b - a = -(kn) = (-k)n, di mana -k juga bulat. Jadi b ≡ a (mod n) (sifat simetris).

(iii) ambil sebarang a,b, dan c ∈ Z.

Jika a ≡ b (mod n) dan b ≡ c (mod n) maka pastilah ada bilangan bulat k dan l sedemikian sehingga a - b = kn dan b - c = ln.

Jadi a - c = (a - b) + (b - c) = kn + ln = (k+l)n, di mana (k+l) juga bilangan bulat. Jadi a ≡ c (mod n) (sifat transitif).■

3.3.4. Definisi Klas Ekuivalensi :

Jika $\sim$ adalah sebuah relasi ekuivalensi pada himpunan S, dan $a \in S$, maka himpunan $[a] = \{ x \in S : a \sim x \}$ disebut klas ekuivalensi yang memuat a.

Bila $a \in Z$, maka $[a] = \{x \in Z : a \equiv x \pmod{n}\}$ adalah klas ekuivalensi yang memuat a.

Contoh 3.3 : Jika n = 12, maka akan didapat :

$1 \equiv 13 \pmod{12}$, $1 \equiv 25 \pmod{12}$,

$1 \equiv -11 \pmod{12}$ dan

$[1] = \{ \ldots, -23, -11, 1, 13, 25, \ldots\}$

$= \{ 1 + 12k : k \in Z \}$

3.3.5. Lemma :

Jika $\sim$ merupakan relasi ekuivalensi pada himpunan S, maka :

(i) $a \in [a]$, untuk setiap $a \in S$

(ii) $[a] = [b]$ bila dan hanya bila $a \sim b$ dan

(iii) jika $[a] \neq [b]$, maka $[a] \cap [b] = \{ \}$.

Bukti :

(i) $a \in [a]$, karena $a \sim a$. (sifat refleksif pada relasi equivalensi)

(ii) Andaikan $[a] = [b]$. Karena $a \in [a]$, maka $a \in [b]$, sehingga $b \sim a$ (menurut definisi 3.3.4). Dengan menggunakan sifat simetris didapat $a \sim b$.

Sebaliknya andaikan $a \sim b$. Maka dengan sifat simetris diperoleh $b \sim a$. Ambil $x \in [a]$, maka $a \sim x$ (menurut definisi 3.3.4). Dengan menggunakaan sifat transitif didapat $b \sim x$.

Jadi $x \in [b]$, sehingga $[a] \subseteq [b]$ .....(1)

Ambil $y \in [b]$, maka $b \sim y$ (menurut definisi 3.3.4) dengan menggunakan sifat transitif di dapat $a \sim y$.

Jadi $y \in [a]$ sehingga $[b] \subseteq [a]$ .....(2)

Dari (1) dan (2) didapat $[a] = [b]$.

(iii) Dibuktikan kontrapositifnya.

Andaikan $[a] \cap [b] \neq \{\ \}$, maka ada $x \in [a] \cap [b]$.

Akibatnya $a \sim x$ dan $b \sim x$, dengan menggunakan sifat simetris diperoleh $x \sim b$ dan dengan sifat transitif diperoleh $a \sim b$. Dengan menggunakan bagian (ii) di atas, kita dapatkan $[a] = [b]$. ∎

Ada banyak cara untuk menyatakan suatu klas ekuivalensi. Sebagai contoh, dalam modulo n, klas ekuivalensi [0] dapat juga dinyatakan dengan [n] atau [-n] atau [100n]. Biasanya klas ekuivalensi [i], dinyatakan dengan memakai bilangan bulat non negatif terkecil yang kongruen dengan i modulo n. Kita dapat menemukan bilangan bulat tersebut dengan membagi i dengan n sebagai berikut:

$$i = qn + r \quad \text{dengan } 0 \leq r < n$$

sehingga [i] =[r].

3.3.6. Definisi:

Jika $i = qn + r$ dengan $0 \leq r < n$, maka r disebut sisa non negatif terkecil dari i modulo n.

Proses ini juga memperlihatkan bahwa setiap bilangan bulat i terdapat dalam salah satu klas ekuivalensi [0], [1], ... [n-1] modulo n. Lagipula tidak ada dua klas ekuivalensi yang sama, sebab jika [i] = [j] maka $i \equiv j \pmod{n}$ (menurut lemma 3.3.5 bagian ii), sehingga $i = j$ (karena $0 \leq i, j < n$).

Contoh 3.4: Dalam modulo 12 terdapat 12 klas ekuivalensi, yaitu :

[0] = { ..., -24 , -12 , 0 , 12 , 24 ,... }

[1] = { ..., -23 , -11 , 1 , 13 , 25 ,... }

[2] = { ..., -22 , -10 , 2 , 14 , 26 ,... }

. . .

[11] = { ..., -25 , -13 ,-1 , 11 , 23 ,... }

3.3.7. Definisi Bilangan bulat modulo n :

Klas-klas ekuivalensi [0] , [1] , ... , [n-1] disebut bilangan-bilangan bulat modulo n. Himpunan {[0], [1], ... , [n-1]} biasanya dilambangkan dengan Z_n.

3.3.8. Lemma :

Jika a ≡ b (mod n) dan c ≡ d (mod n), maka :

(i) a + c ≡ b + d (mod n)

(ii) a - c ≡ b - d (mod n)

(iii) ac ≡ bd (mod n).

Bukti :

Diberikan a ≡ b (mod n) dan c ≡ d (mod n). Maka pastilah ada bilangan bulat i dan j sedemikian sehingga a = b +in dan c = d + jn. Sehingga:

(i) a + c = b + in + d + jn

= b + d + (i+j)n

Jadi $(a + c) - (b+d) = (i + j)n$.

Maka $a+c \equiv b+d \pmod n$.

(ii) $a - c = b + in - (d + jn)$

$= b - d + (i-j)n$

Jadi $(a - c) - (b-d) = (i - j)n$.

Maka $a-c \equiv b-d \pmod n$.

(iii) $ac = (b+in)(d+jn)$

$= bd + bjn + ind + ijnn$

$= bd + (bj + id + ijn)n$.

Jadi $(ac - bd)$ adalah kelipatan n.

Maka $ac \equiv bd \pmod n$. ∎

Dengan demikian kita dapat mendefinisikan operasi aritmetika pada bilangan - bilangan bulat modulo n sebagai berikut :

$[a] + [b] = [\, a + b \,]$

$[a] - [b] = [\, a-b \,]$

$[a] \cdot [b] = [\, ab]$

$[a]^k = [a^k]$, di mana k bilangan bulat positif.

Perlu kita periksa, apakah definisi tersebut "well defined". Cukup kita periksa apakah penjumlahan dan perkalian adalah well defined, karena pengurangan dapat di-

anggap sebagai penjumlahan dengan bilangan negatif dan perpangkatan adalah perkalian yang diulang - ulang.

Membuktikan bahwa operasi penjumlahan adalah well defined berarti membuktikan bahwa jika diambil sebarang $x \in [a]$ dan sebarang $y \in [b]$ maka $[x+y] = [a+b]$.

Bukti:

ambil sebarang $x \in [a]$ dan sebarang $y \in [b]$; maka

$$a \equiv x \pmod{n} \text{ dan } b \equiv y \pmod{n}.$$

Dengan menggunakan lemma 3.3.8 bagian (i) diperoleh :

$$a + b \equiv x + y \pmod{n}.$$

Jadi $[a+b] = [x+y]$ (lemma 3.3.5 bagian ii).

Membuktikan bahwa operasi perkalian adalah well defined berarti membuktikan bahwa jika diambil sebarang $x \in [a]$ dan sebarang $y \in [b]$ maka $[xy] = [ab]$.

Bukti:

ambil sebarang $x \in [a]$ dan sebarang $y \in [b]$; maka

$$a \equiv x \pmod{n} \text{ dan } b \equiv y \pmod{n}.$$

Dengan menggunakan lemma 3.3.8 bagian (iii) diperoleh :

$$ab \equiv xy \pmod{n}.$$

Jadi $[ab] = [xy]$ (lemma 3.3.5 bagian ii).

Jadi operasi penjumlahan dan perkalian tersebut di atas well defined. ∎

3.3.9. Lemma :

Jika ab ≡ cd (mod n) dan a ≡ c (mod n), maka b ≡ d (mod n) asalkan fpb(a,n)=1.

Bukti :

Karena ab ≡ cd (mod n), maka pastilah ada bilangan bulat i sedemikian sehingga ab-cd = in , dan

karena a ≡ c (mod n) maka pastilah ada bilangan bulat j sedemikian sehingga a - c = jn atau c = a - jn.

Maka ab - (a-jn) d = in

ab - ad = in - jdn

a(b-d) = (i-jd)n.

Jadi a(b-d) adalah kelipatan n .

Jika fpb(a,n) = 1, yaitu a dan n tidak mempunyai faktor persekutuan kecuali 1, maka n harus merupakan faktor dari b - d. Jadi b ≡ d (mod n). ∎

3.3.10. Definisi Invers Perkalian :

Bilangan bulat [a] modulo n dikatakan mempunyai invers perkalian jika ada bilangan bulat [b] modulo n sedemikian sehingga [a].[b] = [1].

Jadi [b] di sini berperan sebagai "1/[a]" dan disebut invers perkalian dari [a]. Jika [a].[b] = [1] maka [b].[a] = [1] dan dengan demikian [a] juga merupakan invers perkalian dari [b].

Demikian pula jika a dan b adalah bilangan bulat dengan $0 < a,b < n$ sedemikian sehingga $ab \equiv 1 \pmod{n}$ maka dikatakan bahwa a dan b merupakan invers perkalian satu sama lain.

3.3.11. Corollary

Bilangan bulat [a] modulo n mempunyai invers perkalian bila dan hanya bila fpb(a,n) = 1.

Bukti :

(i) Jika [a].[x] = [1] , maka ax= 1 + kn, di mana k adalah suatu bilangan bulat. Jadi setiap faktor persekutuan dari a dan n juga merupakan faktor dari 1. Maka fpb(a,n)=1.

(ii) Jika fpb(a,n)=1, maka menurut corollary 3.2.4. ada bilangan bulat x dan y sedemikian sehingga 1 = xa + yn. Jadi $1 \equiv xa \pmod{n}$, sehingga [1] = [xa] = [x].[a] . Jadi [x] merupakan invers perkalian dari [a].

D. Bilangan Prima

3.4.1. Definisi Bilangan Prima :

Bilangan bulat positif $p \neq 1$ yang memenuhi sifat: jika $a|p$, maka $a = \pm 1$ atau $a = \pm p$, disebut bilangan prima.

Dua bilangan bulat a dan b dikatakan prima relatif jika fpb(a,b) = 1. Jadi suatu bilangan bulat [a] mempunyai invers perkalian modulo n bila dan hanya bila a dan n prima relatif.

Contoh 3.5: Bilangan - bilangan 1,3,5,dan 7 adalah prima relatif terhadap 8. Jadi [1],[3],[5],dan [7] mempunyai invers perkalian dalam Z_8. Perhatikan tabel berikut:

.	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
[2]	[0]	[2]	[4]	[6]	[0]	[2]	[4]	[6]
[3]	[0]	[3]	[6]	[1]	[4]	[7]	[2]	[5]
[4]	[0]	[4]	[0]	[4]	[0]	[4]	[0]	[4]
[5]	[0]	[5]	[2]	[7]	[4]	[1]	[6]	[3]
[6]	[0]	[6]	[4]	[2]	[0]	[6]	[3]	[2]
[7]	[0]	[7]	[6]	[5]	[4]	[3]	[2]	[1]

tabel 3.1 Tabel operasi . dalam Z_8

Masing - masing dari [1],[3],[5], dan [7] mempunyai

invers perkalian, yaitu dirinya sendiri:

$$[1].[1] = [1]$$

$$[3].[3] = [1]$$

$$[5].[5] = [1]$$

$$[7].[7] = [1]$$

Jika n prima maka semua anggota Z_n mempunyai invers perkalian. Jika n tidak prima maka hanya bilangan yang prima relatif dengan n yang mempunyai invers perkalian.

3.4.2. Definisi Bilangan Komposit :

Setiap bilangan bulat positif $\neq 1$ yang bukan bilangan prima disebut bilangan komposit.

3.4.3. Theorema Fundamental Aritmatika:

Setiap bilangan komposit pasti dapat dinyatakan secara tunggal sebagai perkalian dari faktor-faktor prima.

Bukti :

Bukti dari Theorema Fundamental Aritmatika ini terdiri dari dua bagian. Pada bagian pertama akan diperlihatkan bahwa setiap bilangan komposit dapat dinyatakan seba-

gai perkalian dari faktor-faktor prima, dan pada bagian kedua akan dibuktikan bahwa perkalian dari faktor -faktor prima itu tunggal.

Pembuktian bagian pertama adalah sebagai berikut : Kita coba membagi bilangan itu dengan faktor prima yang terkecil yaitu: 2. Jika bilangan itu habis dibagi 2, maka kita coba ulangi lagi cara ini terhadap hasil baginya, demikian seterusnya sampai hasil baginya tidak habis dibagi 2. Jika bilangan itu atau hasil bagi yang terakhir dari bilangan itu tidak habis dibagi 2, maka kita coba membagi bilangan itu atau hasil baginya yang terakhir dengan bilangan prima berikutnya, yaitu 3. Dengan menggunakan cara yang sama seperti yang di atas, lakukan terus cara ini dengan bilangan prima selanjutnya yaitu 5,7,11,.... Kita tidak perlu mencoba membagi dengan bilangan prima yang lebih besar daripada akar kuadrat bilangan itu. Proses ini berhenti jika akhirnya diperoleh 1 sebagai hasil baginya.
Bilangan-bilangan pembagi itulah yang merupakan faktor-faktor prima dari bilangan tersebut.

Pada bagian kedua ini akan diperlihatkan bahwa

perkalian faktor-faktor prima dari suatu bilangan komposit adalah tunggal.

Diandaikan bahwa bentuk perkalian dari faktor-faktor prima suatu bilangan komposit tidak tunggal. Misalkan ada dua macam bentuk perkalian (yang tidak sama) dari faktor-faktor prima suatu bilangan komposit.

Perkalian dari faktor-faktor prima itu dapat disusun menurut urutan besarnya, mulai dari yang terkecil, sebagai berikut: 2 dulu, jika ada , diikuti dengan 3, jika ada, dan seterusnya. Perkalian faktor-faktor prima itu dapat kita tulis sebagai berikut :

$P \times Q \times R \times S \times \ldots \times W = P' \times Q' \times R' \times S' \times \ldots \times W'$

Karena P merupakan faktor dari bilangan yang dinyatakan pada ruas kiri, maka P juga merupakan faktor dari ruas kanan. Karena itu P harus sama dengan salah satu dari faktor-faktor P', Q', R', S',..., W'. Karena faktor-faktor itu telah di susun terurut mulai dari faktor prima yang terkecil hingga faktor prima terbesar dari kiri ke kanan, maka P tidak mungkin lebih kecil dari P'. Di lain pihak, dengan cara yang sama akan didapat bahwa P' adalah sama dengan salah satu dari faktor - faktor P, Q, R, S, ..., W sehingga P' tidak mungkin lebih kecil dari P. Jadi P = P'.

Dengan membagi persamaan di atas dengan P, kita mendapatkan :

$$Q \times R \times S \times \ldots \times W = Q' \times R' \times S' \times \ldots \times W'$$

Dengan cara yang sama seperti di atas akan didapat $Q=Q'$. Demikian cara ini diteruskan untuk setiap faktor ruas kiri, yang ternyata sama dengan faktor bersesuaian di ruas kanan.

Jadi pada akhirnya akan didapat bahwa kedua bentuk perkalian faktor-faktor prima itu adalah sama.

Terjadi kontradiksi. Jadi hanya ada tunggal bentuk perkalian faktor-faktor prima dari sebuah bilangan komposit .

3.4.4. Theorema:

Ada tak berhingga banyak bilangan prima.

Bukti :

Andaikan bahwa ada berhingga banyak bilangan prima. Karena ada berhingga banyak bilangan prima, maka pastilah ada bilangan prima yang terbesar, misalkan p. Didefinisikan N sebagai perkalian semua bilangan prima dari 2 sampai dengan p. Karena $N + 1 > p$ dan p adalah adalah bilangan prima yang terbesar, maka $N + 1$ adalah bilangan komposit. Menurut Theorema Fundamental Aritme-

tika pastilah N + 1 dapat dinyatakan sebagai perkalian dari faktor-faktor prima. Dari caranya membentuk bilangan N + 1 itu jelaslah bahwa jika N + 1 dibagi dengan bilangan-bilangan prima dari 2 sampai p, pastilah akan menghasilkan sisa 1. Jadi faktor-faktor primanya lebih besar dari p. Dengan demikian ada bilangan prima yang lebih besar dari p. Terjadi kontradiksi.

Maka ada tak berhingga banyak bilangan prima. ∎

3.4.5. Theorema :

Untuk sebarang $a,b \in Z^+$ dan sebarang bilangan prima p berlaku: jika $p|ab$ maka $p|a$ atau $p|b$.

Bukti :

Andaikan p bukan faktor dari a, maka akan dibuktikan bahwa $p|b$.

Karena p prima dan p bukan faktor dari a, maka fpb(p,a)=1. Menurut corollary 3.2.4 :

$$1 = xp + ya \text{ di mana } x,y \in Z.$$

Jika kedua ruas dari $1 = xp + ya$ dikalikan dengan b, maka didapat : $b = xpb + yab$.

Jelas bahwa $p|xpb$ dan $p|yab$ (karena $p|ab$).

Jadi $p|b$. ∎

3.4.6. Theorema :

Jika p prima dan fpb(a,p) = 1, maka $a^{p-1} \equiv 1 \pmod{p}$.

Bukti:

Karena p prima, maka untuk sebarang bilangan bulat n, fpb(n,p) adalah 1 atau p. Jadi fpb(n,p) = 1 bila dan hanya bila n bukan kelipatan p, yaitu $n \not\equiv 0 \pmod{p}$. Jadi yang kita asumsikan adalah $a \not\equiv 0 \pmod{p}$.

Perhatikan klas-klas ekuivalensi berikut ini :

[a],[2a],[3a],...,[(p-1)a]. .....(B)

Akan diperlihatkan bahwa:

(i) tidak ada diantara klas-klas ekuivalensi tersebut yang sama dengan [0], dan

(ii) tidak ada dua klas ekuivalensi yang sama.

(i) Dengan kontradiksi.

Diandaikan bahwa [na] = [0], di mana $1 \leq n \leq (p-1)$.

Maka $na \equiv 0 \pmod{p}$. (Menurut lemma 3.3.5 bagian ii).

Jadi p merupakan faktor dari na dan karena fpb(a,p) = 1, maka menurut theorema 3.4.5, p merupakan faktor dari n.

Terjadi kontradiksi, karena $1 \leq n \leq (p-1)$.

Jadi tidak ada di antara klas-klas ekuivalensi tersebut yang sama dengan [0].

(ii) Dengan kontradiksi.

Diandaikan bahwa [na] = [ma] dimana $1 \leq n,m \leq (p-1)$.

Maka $na \equiv ma \pmod p$ (menurut lemma 3.3.5.bagian ii).

Karena fpb(a,p) = 1, maka menurut lemma 3.3.9 :

$n \equiv m \pmod p$.

Terjadi kontradiksi, karena n dan m adalah bilangan-bilangan bulat positif yang lebih kecil dari p.

Jadi (p-1) klas ekuivalensi tersebut di atas adalah sama dengan klas-klas ekuivalensi [1], [2], ...,[p-1]. Maka :

$[a].[2a].[3a]. \ldots .[(p-1)a] = [1].[2].[3]. \ldots .[p-1]$

$[a(2a)\ldots(p-1)a] = [(p-1)!]$

$a(2a) \ldots ((p-1)a) \equiv (p-1)! \pmod p$ lemma 3.3.5.ii

$a^{p-1}(p-1)! \equiv (p-1)! \pmod p$

$a^{p-1} \equiv 1 \pmod p$ lemma 3.3.9

karena fpb(p,(p-1)!) = 1. ∎

3.4.7. Lemma :

Jika fpb(a,n) = fpb(b,n) = 1, maka fpb(ab,n) = 1.

Bukti :

Dibuktikan dengan menggunakan kontrapositifnya.

Diandaikan bahwa fpb(ab,n) = d > 1.

Jika d prima, maka d mempunyai faktor prima, yaitu dirinya sendiri.

Jika d komposit, maka menurut Theorema Fundamental Aritmetika, pastilah d punya faktor prima.

Andaikan bahwa salah satu faktor prima d adalah p.

Maka p merupakan faktor dari ab dan n. Menurut theorema 3.4.5, p merupakan faktor dari a atau dari b.

Jadi fpb(a,n) $\neq$ 1 atau fpb(b,n) $\neq$ 1. ∎

3.4.8. Theorema:

Jika n=pq di mana p dan q adalah bilangan prima yang tidak sama dan a adalah bilangan bulat positif sedemkian sehingga fpb(a,n) = 1, maka $a^{(p-1)(q-1)} \equiv 1 \pmod{n}$.

Bukti:

Andaikan $Z_n = \{[0],[1],[2], \ldots [n-1]\}$.

Didefinisikan himpunan A:

$$A = \{[x] \in Z_n \mid fpb(x,n) = 1 \}.$$

Akan dihitung banyaknya anggota himpunan A dengan menghitung banyaknya anggota himpunan $Z_n - A$.

Untuk sebarang $[i] \in Z_n$, kemungkinan dari fpb(i,n) adalah 1, p, q, atau pq.

Satu-satunya [x] dengan fpb(x,n) = pq adalah [x] = [0].

Elemen-elemen [x] dengan fpb(x,n) = p adalah

[p], [2p], [3p], ... , [(q-1)p].

Elemen-elemen [x] dengan fpb(x,n) = q adalah

[q], [2q], [3q], ... , [(p-1)q].

Perhatikan bahwa bilangan-bilangan bulat modulo n tersebut tidak ada yang sama. Sebab jika ip = jq dengan $1 \leq i \leq (q-1)$ dan $1 \leq j \leq (p-1)$, maka p merupakan faktor dari jq. Karena p dan q bilangan prima dan $p \neq q$ maka p merupakan faktor dari j.

Terjadi kontradiksi, karena $j \leq (p-1)$.

Jadi bilangan-bilangan bulat modulo n

[p], ... , [(q-1)p], [q], ... , [(p-1)q] semuanya berbeda. Dengan demikian himpunan Z_n - A memuat 1+ (q-1) + (p-1) buah anggota. Jadi himpunan A mempunyai n - 1 - (p-1) - (q-1) = pq - p - q + 1 = (p-1) (q-1) buah elemen.

Misalkan $A = \{[r_1],[r_2], \ldots , [r_s]\}$,

di mana $0 < r_1 < r_2 < \ldots < r_s < n$ dan $s = (p-1)(q-1)$.

Ambil sebarang bilangan bulat positif a sedemikian sehingga fpb(a,n) = 1 dan perhatikan himpunan

$S = \{[ar_1],[ar_2], \ldots [ar_s]\}$. Dengan lemma 3.4.7 diperoleh $fpb(ar_i,n)=1$, untuk $i = 1,2,\ldots,s$. Dan juga

fpb($r_1r_2 \ldots r_s$,n) = 1.....(*)

Akan diperlihatkan bahwa S = A.

Karena fpb(ar_i,n) = 1, maka $[ar_i] \in A$. Jadi $[ar_i] = [r_j]$ untuk suatu j. Lagi pula tidak ada dua buah elemen dalam himpunan S yang sama.

Andaikan $[ar_i] = [ar_k]$ untuk suatu nilai i dan k dengan $r_i \neq r_k$, maka:

$$ar_i \equiv ar_k \pmod{n}$$

$$r_i \equiv r_k \pmod{n}$$

$r_i = r_k$, sebab $r_i, r_k < n$.

Terjadi kontradiksi.

Jadi S = A. Maka

$$[r_1].[r_2]. \ldots .[r_s] = [ar_1].[ar_2]. \ldots .[ar_s].$$

Jadi $r_1r_2 \ldots r_s \equiv ar_1ar_2 \ldots ar_s \pmod{n}$ lemma 3.3.5.

$$r_1r_2 \ldots r_s \equiv a^s r_1r_2 \ldots r_s \pmod{n}$$

$$1 \equiv a^s \pmod{n}$$ lemma 3.3.9 dan (*)

$$\equiv a^{(p-1)(q-1)} \pmod{n}.$$ ∎

E. Algoritma Fastexp

Algoritma Fastexp adalah suatu algoritma yang dirancang untuk membantu mempercepat suatu perhitungan bilangan eksponen yang besar. Dasar dari Algoritma Fast-

exp ini adalah :

3.5.1. Theorema :

Terhadap suatu modulo tertentu berlaku bahwa sisa dari kuadrat suatu bilangan bulat kongruen dengan kuadrat sisa dari bilangan itu.

Bukti:

Andaikan modulonya adalah $n \in Z^+$. Ambil sebarang bilangan bulat $x > n$. Maka

$x = kn + t$, di mana $k \in Z$, $0 \leq t < n$, dan

$x^2 = mn + s$, di mana $m \in Z$, $0 \leq s < n$.

Jadi : $t^2 - s = (x-kn)^2 - (x^2-mn)$

$= x^2 + k^2n^2 - 2xkn - x^2 + mn$

$= k^2n^2 - 2xkn + mn$

$= (k^2n - 2xk + m)\, n$

Terbukti $s \equiv t^2 \pmod n$. ∎

Dengan cara biasa (manual), kita dapat menggunakan theorema ini secara berulang-ulang.

Contoh 3.7: Untuk menghitung $7^{89} \pmod{13}$, kita hitung dulu nilai $7^2 \pmod{13}$ dari $7 \pmod{13}$, kemudian kita hitung $7^4 \pmod{13}$ dari $7^2 \pmod{13}$, lalu

$7^8 \pmod{13}$ dari $7^4 \pmod{13}$,dan seterusnya sampai $7^{64} \pmod{13}$ dari $7^{32} \pmod{13}$. Kita tidak perlu menghitung melebihi bilangan yang akan kita hitung, yaitu $7^{89} \pmod{13}$.

$$7^2 \pmod{13} \equiv (7)^2 \pmod{13} \equiv 10 \pmod{13}$$
$$7^4 \pmod{13} \equiv (10)^2 \pmod{13} \equiv 9 \pmod{13}$$
$$7^8 \pmod{13} \equiv (9)^2 \pmod{13} \equiv 3 \pmod{13}$$
$$7^{16} \pmod{13} \equiv (3)^2 \pmod{13} \equiv 9 \pmod{13}$$
$$7^{32} \pmod{13} \equiv (9)^2 \pmod{13} \equiv 3 \pmod{13}$$
$$7^{64} \pmod{13} \equiv (3)^2 \pmod{13} \equiv 9 \pmod{13}$$

Padahal $7^{89} \pmod{13} \equiv$

$$7^{64} \pmod{13} \times 7^{16} \pmod{13} \times 7^8 \pmod{13} \times 7^1 \pmod{13}$$
$$\equiv 9 \pmod{13} \times 9 \pmod{13} \times 3 \pmod{13} \times 7 \pmod{13}$$
$$\equiv 3 \pmod{13} \times 3 \pmod{13} \times 7 \pmod{13}$$
$$\equiv 9 \pmod{13} \times 7 \pmod{13}$$
$$\equiv 11 \pmod{13}.$$

Akan lebih baik jika kita mempunyai sebuah program yang dapat mengerjakan perhitungan ini[1], berikut akan

---

[1] Lihat lampiran B, hal. 73

disajikan Algoritma Fastexp yang dapat digunakan untuk membuat sebuah program :

Algoritma Fastexp [2]:

```
Step 0   Input x,m, set ans: = 1
Step 2   Divide m by 2 to obstain quotiont q and remain
         der r
Step 2,5 If r = 1 , set ans: = ans * x
Step 3   If q = 0, then stop
Step 4   Set m: = q
Step 5.5 Set x: = x * x
Step 6   Go to step 2
```

Menurut George Mackiw :

Dengan Algoritma Fastexp ini, banyaknya perkalian yang perlu dikerjakan untuk menghitung R^a tidak lebih dari $2\ {}^2\log a$. Jadi kalaupun a adalah suatu bilangan dengan 200 digit, banyaknya perkalian yang harus dikerjakan dengan Algoritma Fastexp ini paling banyak adalah $2\ {}^2\log 10^{200} = 400\ {}^2\log 10 \approx 1.330$ perkalian. Sedangkan dengan cara biasa kita akan melakukan 10^{200} perkalian.[3]

---

[2] Michael O Albertson dan Joan P Hutchinson [1988], hal 91

[3] Greorge Mackiw [1985], hlm.129.

BAB IV

SISTEM PENYANDIAN RSA

Sistem penyandian RSA yang akan disajikan dalam bab ini didasarkan atas teori bilangan, khususnya algoritma Euclides, bilangan modulo, dan bilangan prima.

Pada tahun 1977 Ronald L Rivest dari Massachusetts Institute of Technology, bersama-sama dengan Adi Shamir dari Weizmann Institute of Science dan Leonard M. Adleman dari University of Southern California merancang suatu sistem penyandian yang tidak mudah terpecahkan oleh para ahli cryptanalysis. Jika kita tidak mengetahui parameter kuncinya, yaitu dua buah bilangan prima dengan digit yang sangat besar, katakanlah keduanya 100 digit, maka akan dibutuhkan waktu selama jutaan tahun untuk memecahkan suatu pesan sandi yang menggunakan sistem penyandian dari Rivest, Shamir dan Adleman ini.[4]

Sistem penyandian RSA ini didasarkan pada kenyataan bahwa sangat sulit untuk memfaktorkan suatu bilangan komposit yang merupakan perkalian dari dua bilangan prima yang sangat besar, misalnya keduanya berdigit 100.

---

[4] Martin Gardner [1977], hlm 123

Tentu lebih mudah menentukan dua buah bilangan prima dan kemudian mengalikannya dari pada menentukan faktor-faktor prima dari suatu bilangan komposit yang berdigit 200.

Keistimewaan sistem penyandian ini adalah bahwa pengirim dan penerima pesan hanya perlu merahasiakan dua buah bilangan prima saja. Jadi pengirim dan penerima pesan tidak perlu mengkhawatirkan jalur komunikasi yang semakin tidak aman untuk mengirimkan pesan tersebut.

A. Langkah Awal Penggunaan Sistem Penyandian

Pengirim dan penerima pesan harus telah memilih dan menyepakati dua buah bilangan prima p dan q secara acak, kemudian hitung n = pq. Pemilihan bilangan-bilangan prima tersebut dapat didasarkan pada seberapa pentingnya pesan yang dikirimkan ini bagi pihak lawan atau terhadap ahli cryptanalysis. Artinya jika pesan yang dikirimkan itu sangat penting maka sipemakai dapat memilih bilangan prima yang berdigit besar umpamanya berdigit 100 dan jika pesan yang dikirimkan itu tidak sangat penting, maka sipemakai dapat memilih bilangan prima dengan digit yang lebih kecil misalnya berdigit 50.

Selanjutnya dipilih sebarang bilangan bulat a yang memenuhi $0 < a < (p-1)(q-1)$, sedemikian sehingga $fpb(a,(p-1)(q-1)) = 1$. Bilangan a ini disebut bilangan eksponen. Menurut corollary 3.3.11, bilangan bulat a modulo $(p-1)(q-1)$ akan mempunyai invers perkalian bila dan hanya bila $fpb(a,(p-1)(q-1)) = 1$.

Yang penting pada langkah awal ini adalah merahasikan bilangan p dan q,sebab jika bilangan ini dapat diketahui oleh ahli cryptanalysis, maka dengan mudah isi pesan sandi tersebut akan diketahui.

B. Menyandikan Pesan

Memproses suatu pesan yang masih berupa kalimat biasa menjadi pesan sandi disebut menyandikan pesan atau encryption.

Jika ada pesan yang akan dikirimkan, maka pengirim harus mengganti setiap huruf dari pesan tersebut dengan suatu angka desimal. Pengubahan dari bentuk kalimat biasa menjadi bentuk angka desimal ini dapat dilakukan dengan bermacam-macam cara, tergantung pada kesepakatan antara pengirim dan penerima pesan, misalnya dengan memakai suatu tabel yang dibuat bersama antara pengirim

dan penerima pesan, atau menggunakan tabel ASCII (The American Standard Code for Information Interchange) [5], atau dapat juga dengan menggunakan tabel lain seperti yang terdapat dalam majalah Scientific American edisi Agustus 1979 berikut ini:

a = 00	b = 01	c = 02	d = 03	e = 04	f = 05
g = 06	h = 07	i = 08	j = 09	k = 10	l = 11
m = 12	n = 13	o = 14	p = 15	q = 16	r = 17
s = 18	t = 19	u = 20	v = 21	w = 22	x = 23
y = 24	z = 25	A = 26	B = 27	C = 28	D = 29
E = 30	F = 31	G = 32	H = 33	I = 34	J = 35
K = 36	L = 37	M = 38	N = 39	O = 40	P = 41
Q = 42	R = 43	S = 44	T = 45	U = 46	V = 47
W = 48	X = 49	Y = 50	Z = 51	0 = 52	1 = 53
2 = 54	3 = 55	4 = 56	5 = 57	6 = 58	7 = 59
8 = 60	9 = 61	= 62	. = 63	, = 64	; = 65
? = 66	...	...			

tabel 4.1 Tabel Penyandian

Contoh 4.1 : Pesan "IKIP Sanata Dharma YOGYAKARTA" diubah ke bentuk angka desimal dengan menggunakan tabel ASCII menjadi :

737573803283971109711697 32...

di mana : I menjadi 73

K menjadi 75

P menjadi 80, dan seterusnya.

---

5 Lihat Appendiks A, hal. 72.

Contoh 4.2 : Pesan "IKIP Sanata Dharma YOGYAKARTA" diubah ke bentuk angka desimal dengan menggunakan tabel 4.1 menjadi : 343634416244001300 1900...

di mana: I menjadi 34

K menjadi 36

P menjadi 41, dan seterusnya.

Pesan yang telah diubah menjadi bentuk angka desimal itu dapat dibagi-bagi ke dalam blok-blok, masing-masing dengan panjang B digit, sehingga pesan tersebut menjadi barisan bilangan bulat M_i dengan $0 < M_i < n$ untuk $i = 1,2,\ldots,k$. Jika sipemakai mempunyai sarana yang memadai untuk mengerjakan suatu perhitungan, maka ia dapat saja memilih blok yang cukup panjang, misalnya 5, 6, ... digit. Tetapi jika sipemakai hanya mempunyai sarana yang sederhana saja, misalnya sebuah kalkulator atau komputer dengan kapasitas yang kecil maka ia dapat memilih blok dengan panjang 2,3 atau 4 digit. Pertimbangan di atas tidaklah bersifat mutlak, tetapi sepenuhnya diserahkan kepada pemakai sistem penyandian ini.

Pesan sandi yang akan dikirimkan adalah barisan bilangan bulat R_i yang dikacaukan dengan perhitungan:

$$R_i \equiv M_i^{\,a} \pmod{n} \quad \text{di mana } 0 < R_i < n.$$

yang ekuivalen dengan

$$M_i^a = Qn + R_i \text{ di mana } 0 < R_i < n.$$

Jadi R_i adalah bilangan positif < n yang merupakan sisa bilangan M_i^a dibagi dengan n.

Agar nanti (dalam proses pembacaan pesan sandi) kita dapat memakai theorema 3.4.8, bilangan M_i, tersebut harus memenuhi syarat $fpb(M_i,n) = 1$.

Biasanya M_i^a adalah bilangan bulat positif yang sangat besar, sehingga tidaklah mudah untuk menghitung R_i. Untuk menghitung R_i itu, algoritma Fastexp (yang dibahas pada bab III) akan sangat membantu.

Selanjutnya (a,n) disebut pasangan kunci umum dalam sistem penyandian RSA.

C. Membaca Pesan Sandi

Memproses suatu pesan sandi menjadi pesan yang berupa kalimat biasa dinamakan membaca pesan atau decryption.

Penerima pesan kini telah menerima pesan sandi dan mempunyai dua buah bilangan prima p dan q, serta sebuah bilangan bulat a yang telah dipilih. Agar penerima pesan

dapat membaca pesan sandi tersebut, ia harus terlebih dahulu mengubah pesan yang telah dikacaukan itu dengan memakai invers dari bilangan eksponen a dalam modulo (p-1)(q-1), namakan d. Bilangan d tersebut adalah bilangan bulat positif sedemikian sehingga

$$\boxed{ad \equiv 1 \pmod{(p-1)(q-1)}.}$$

Bilangan d ini dapat dihitung dengan menggunakan persamaan-persamaan Euclides seperti dalam bukti corollary 3.2.4.

Kita tahu bahwa $R_i \equiv M_i^{\,a} \pmod n$, maka

$$\begin{aligned} R_i^{\,d} &\equiv (M_i^{\,a})^d \pmod n \\ &\equiv M_i^{\,ad} \pmod n \\ &\equiv M_i^{\,1+k(p-1)(q-1)} \pmod n \end{aligned}$$

untuk suatu bilangan bulat k,

karena $ad \equiv 1 \pmod{(p-1)(q-1)}$

$$\begin{aligned} &\equiv M_i (M_i^{(p-1)(q-1)})^k \pmod n \\ &\equiv M_i \, 1^k \pmod n \quad \text{(theorema 3.4.8)} \\ &\equiv M_i \pmod n. \end{aligned}$$

Maka $$\boxed{R_i^{\,d} \equiv M_i \pmod n}$$

Jadi M_i adalah sisa dari $R_i^{\,d}$ jika dibagi dengan n (dapat dicari dengan memakai algoritma Fastexp). Langkah terakhir ialah mengubah barisan

bilangan M_i tersebut menjadi pesan dalam kalimat biasa dengan menggunakan tabel yang sama. Pasangan (d,n) disebut pasangan kunci pembacaan pesan sandi.

D. Diagram Sistem Penyandian RSA

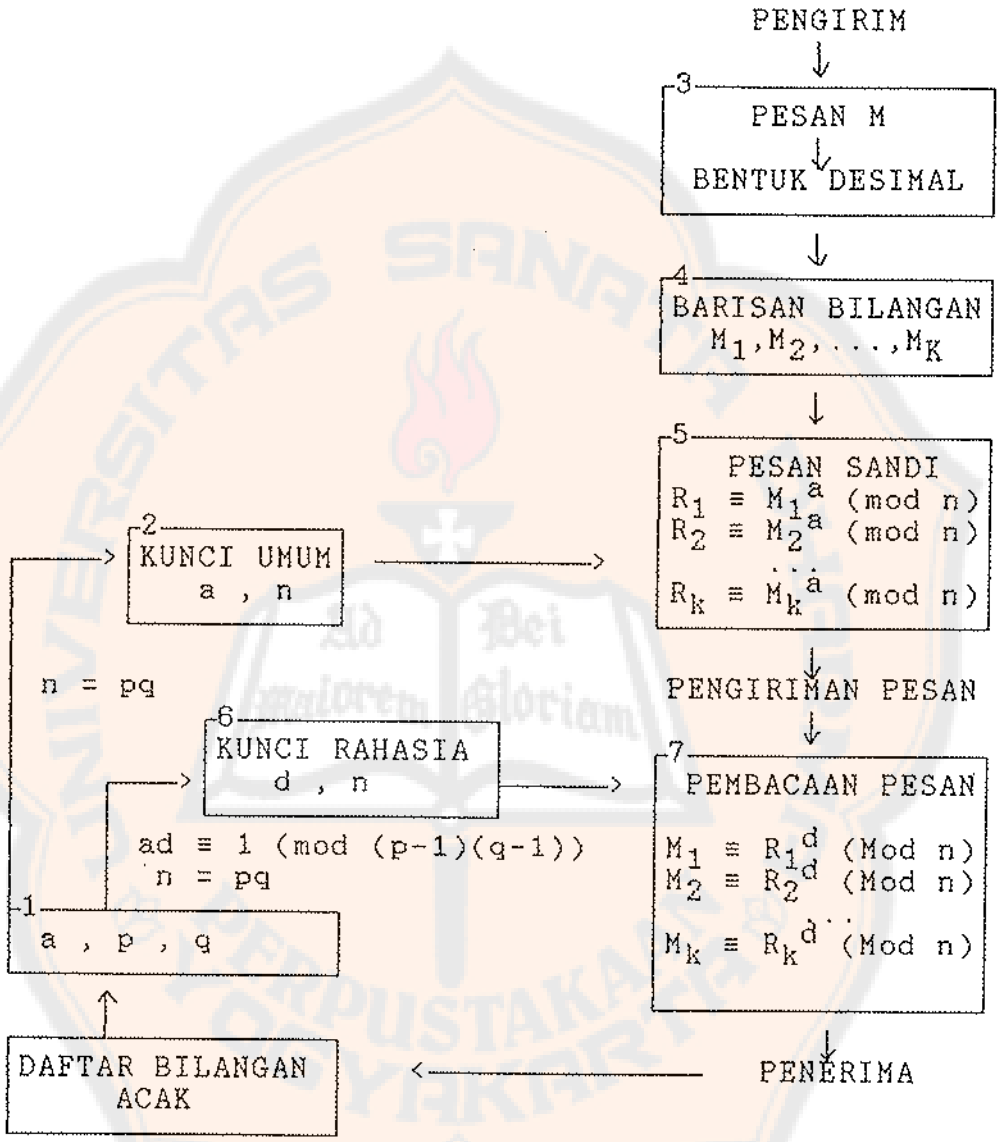


Diagram ini dikutip dari majalah Scientific American Agustus 1979, halaman 138

E. Contoh Penyandian RSA

Pada bagian ini akan disajikan contoh penyandian RSA. Penulis hanya akan memberikan contoh dengan digit yang kecil, yang dapat digunakan sebagai perbandingan jika kita ingin memilih digit yang lebih besar.

Langkah 1 : Memilih dua buah bilangan prima yang tidak sama, misalkan $p = 73$ dan $q = 151$, dan suatu bilangan eksponen a yang prima relatif dengan $(p-1)(q-1) = (73-1)(151-1) = 10800$, misalkan $a = 11$. Dengan menggunakan theorema 3.2.3 kita dapat memeriksa apakah $fpb(11,10800) = 1$:

$$10800 = 981 \cdot 11 + 9$$
$$11 = 1 \cdot 9 + 2$$
$$9 = 4 \cdot 2 + 1$$
$$2 = 2 \cdot 1 + 0$$

Jadi $fpb(a,(p-1)(q-1)) = 1$ yaitu a prima relatif terhadap $(p-1)(q-1)$.

Langkah 2 : Didapat a= 11 dan n = pq = 73 x 151 = 11023. Rahasiakan p dan q yang telah dipilih itu. Selanjutnya nilai a dan n disebut dengan kunci umum, karena tidak perlu dirahasiakan.

Langkah 3, langkah 4 dan langkah 5 dikerjakan oleh pengirim pesan untuk menyandikan pesan yang akan dikirimkannya. Kita andaikan bahwa pesan yang akan dikirimkan adalah " IKIP Sanata Dharma YOGYAKARTA" dan disepakati bahwa pengubahan ke bentuk desimal menggunakan tabel 4.1 yang terdapat dalam BAB IV, dengan panjang blok 4 digit.

Langkah 3 : Pesan "IKIP Sanata Dharma YOGYAKARTA" diubah ke bentuk desimal dengan menggunakan tabel 4.1 menjadi : 343634416244001300190062290700171200625040325026362643452 6

Langkah 4 : Pesan yang telah menjadi bentuk desimal ini dibagi ke dalam blok-blok sepanjang 4 digit, menjadi :

M_1 = 3436 M_2 = 3441 M_3 = 6244

M_4 = 0013 M_5 = 0019 M_6 = 0062

M_7 = 2907 M_8 = 0017 M_9 = 1200

M_{10}= 6250 M_{11}= 4032 M_{12}= 5026

M_{13}= 3626 M_{14}= 4345 M_{15}= 2662

Langkah 5: Pada langkah ini sipengirim, harus memeriksa dulu bahwa fpb(M_i,n) = 1. Jika tidak, maka pesan dinyatakan rusak dan kembali pada langkah 1. Pesan yang dikirimkan adalah barisan bilangan M_i yang telah dikacaukan dengan menggunakan persamaan :

$$R_i \equiv M_i^{a} \text{ (modulo n) dimana } 0 < R_i < n$$

Kita dapat menghitung R_i dengan bantuan Algoritma Fastexp .

$R_1 \equiv M_1^{11} (\text{mod } 11023) \equiv M_1^2 M_1^8 M_1^1 (\text{mod } 11023)$

$M_1^2 \equiv (3436)^2 (\text{mod } 11023) \equiv 463 (\text{mod } 11023)$

$M_1^4 = (M_1^2)^2 \equiv (463)^2 (\text{mod } 11023)$

$\equiv 4932 (\text{mod } 11023)$

$M_1^8 = (M_1^4)^2 \equiv (4932)^2 (\text{mod } 11023)$

$\equiv 7886 (\text{mod } 11023)$

$R_1 \equiv M_1^{11} (\text{mod } 11023)$

$\equiv M_1^2 M_1^8 M_1^1 (\text{mod } 11023)$

$\equiv 463 \cdot 7886 \cdot 3436 (\text{mod } 11023)$

$\equiv 2605 \cdot 3436 (\text{mod } 11023)$

$\equiv 104 (\text{mod } 11023)$

$R_2 \equiv M_2^{11} (\text{mod } 11023) \equiv M_2^2 M_2^8 M_2^1 (\text{mod } 11023)$

$M_2^2 \equiv (3441)^2 \pmod{11023} \equiv 1779 \pmod{11023}$

$M_2^4 = (M_2^2)^2 \equiv (1779)^2 \pmod{11023}$

$\equiv 1240 \pmod{11023}$

$M_2^8 = (M_2^4)^2 \equiv (1240)^2 \pmod{11023}$

$\equiv 5403 \pmod{11023}$

$R_2 \equiv M_2^{11} \pmod{11023}$

$\equiv M_2^2 M_2^8 M_2^1 \pmod{11023}$

$\equiv 1779 \cdot 5403 \cdot 3441 \pmod{11023}$

$\equiv 10904 \cdot 3441 \pmod{11023}$

$\equiv 9395 \pmod{11023}$

$R_3 \equiv M_3^{11} \pmod{11023} \equiv M_3^2 M_3^8 M_3^1 \pmod{11023}$

$M_3^2 \equiv (6244)^2 \pmod{11023} \equiv 10208 \pmod{11023}$

$M_3^4 = (M_3^2)^2 \equiv (10208)^2 \pmod{11023}$

$\equiv 2845 \pmod{11023}$

$M_3^8 = (M_3^4)^2 \equiv (2845)^2 \pmod{11023}$

$\equiv 3143 \pmod{11023}$

$R_3 \equiv M_3^{11} \pmod{11023}$

$\equiv M_3^2 M_3^8 M_3^1 \pmod{11023}$

$\equiv 10208 \cdot 3143 \cdot 6244 \pmod{11023}$

$\equiv 6814 \cdot 6244 \pmod{11023}$

$\equiv$ 8859 (mod 11023)

Demikian seterusnya untuk R_4, R_5,...,R_{15}, didapat :

R_4 = 4385 R_{10} = 10629

R_5 = 7116 R_{11} = 3753

R_6 = 9091 R_{12} = 550

R_7 = 7806 R_{13} = 6427

R_8 = 5891 R_{14} = 6929

R_9 = 294 R_{15} = 9099

Selanjutnya pesan yang telah disandikan tersebut dikirimkan berupa barisan bilangan R_i di mana $0 < R_i < n$ dan $i = 1,2,...,15$.

Pada langkah 6 dan 7 sipenerima pesan mulai bekerja untuk membaca pesan sandi yang telah diterimanya.

Langkah 6: Dengan menggunakan bukti dalam corrolary 3.2.4 dapat dihitung kunci rahasia untuk membaca pesan sandi yang diterima.

$1 = 9 - 4.2$

$1 = 9 - 4(11 - 1.9)$

$1 = 5.9 - 4.11$

$1 = 5(10800 - 981.11) - 4.11$

$1 = 5.10800 - (5.981 + 4).11$

$1 = 5.10800 - (4905 + 4).11$

$1 = 5.10800 - 4909.11$

$1 - (-4909.11) = 5.10800$

$1 \equiv -4909.11 \pmod{10800}$

Jadi -4909 merupakan invers perkalian dari 11 dalam modulo 10800. Tetapi

$-4909 \pmod{10800} \equiv 10800-4909 \pmod{10800}$

$\equiv 5891 \pmod{10800}$.

Jadi d = 5891.

Langkah 7 : Pada tahap ini si penerima dapat membaca pesan sandi dengan menghitung M_i,di mana $M_i \equiv R_i^d \pmod{n}$. Bilangan M_i dapat dihitung dengan bantuan algoritma Fastexp.

$M_1 \equiv R_1^{5891} \pmod{11023}$

$\equiv R_1^{4096} \; R_1^{1024} \; R_1^{512} \; R_1^{256} \; R_1^{2} \; R_1^{1} \pmod{11023}$

$R_1^{1} \equiv 104 \pmod{11023}$

$R_1^{2} = (R_1^{1})^2 \equiv (104)^2 \pmod{11023}$

$\equiv 10816 \pmod{11023}$

$$R_1^4 = (R_1^2)^2 \equiv (10816)^2 \pmod{11023}$$
$$\equiv 9780 \pmod{11023}$$

$$R_1^8 = (R_1^4)^2 \equiv (9780)^2 \pmod{11023}$$
$$\equiv 1829 \pmod{11023}$$

$$R_1^{16} = (R_1^8)^2 \equiv (1829)^2 \pmod{11023}$$
$$\equiv 5272 \pmod{11023}$$

$$R_1^{32} = (R_1^{16})^2 \equiv (5272)^2 \pmod{11023}$$
$$\equiv 5001 \pmod{11023}$$

$$R_1^{64} = (R_1^{32})^2 \equiv (5001)^2 \pmod{11023}$$
$$\equiv 9837 \pmod{11023}$$

$$R_1^{128} = (R_1^{64})^2 \equiv (9837)^2 \pmod{11023}$$
$$\equiv 6675 \pmod{11023}$$

$$R_1^{256} = (R_1^{128})^2 \equiv (6675)^2 \pmod{11023}$$
$$\equiv 659 \pmod{11023}$$

$$R_1^{512} = (R_1^{256})^2 \equiv (659)^2 \pmod{11023}$$
$$\equiv 4384 \pmod{11023}$$

$$R_1^{1024} = (R_1^{512})^2 \equiv (4384)^2 \pmod{11023}$$
$$\equiv 6367 \pmod{11023}$$

$$R_1^{2048} = (R_1^{1024})^2 \equiv (6367)^2 \pmod{11023}$$
$$\equiv 7118 \pmod{11023}$$

$$R_1^{4096} = (R_1^{2048})^2 \equiv (7118)^2 \pmod{11023}$$
$$\equiv 4216 \pmod{11023}$$

$$M_1 \equiv R_1^{5891} \pmod{11023}$$

$\equiv R_1^{4096} \quad R_1^{1024} \quad R_1^{512} \quad R_1^{256} \; R_1^{2} \; R_1^{1}$

(mod 11023)

$\equiv 4216.6367.4384.659.10816.104 \pmod{11023}$

$\equiv 2267.4384.659.10816.104 \pmod{11023}$

$\equiv 6805.659.10816.104 \pmod{11023}$

$\equiv 9157.10816.104 \pmod{11023}$

$\equiv 457.104 \pmod{11023}$

$\equiv 3436 \pmod{11023}$

Demikian seterusnya untuk M_2, M_3,...M_{15}, didapat:

$M_1 = 3436$	$M_2 = 3441$	$M_3 = 6244$
$M_4 = 0013$	$M_5 = 0019$	$M_6 = 0062$
$M_7 = 2907$	$M_8 = 0017$	$M_9 = 1200$
$M_{10} = 6250$	$M_{11} = 4032$	$M_{12} = 5026$
$M_{13} = 3626$	$M_{14} = 4345$	$M_{15} = 2662$

Selanjutnya di kembalikan ke bentuk desimal tanpa menggunakan blok menjadi :

34363441624400130019006229070017120062504032 5026362643452662. Dengan menggunakan tabel 4.1 didapat pesan :

IKIP Sanata Dharma YOGYAKARTA

Demikianlah keseluruhan proses dari sistim penyandian yang telah dirancang oleh 3 ahli matematika yaitu Rivest, Shamir dan Adleman.

BAB V

PEMBAHASAN SISTEM PENYANDIAN RSA

Kekuatan dan sekaligus ide dasar dari Sistem Penyandian RSA ini terletak pada sulitnya memfaktorkan suatu bilangan bulat yang merupakan perkalian dua buah bilangan prima dengan digit yang sangat besar. Sangat sulit untuk menemukan suatu algoritma yang efisien untuk menemukan faktor-faktor prima dari sebuah bilangan komposit dengan digit yang besar sekali.

Jadi perlu diperhatikan bahwa bilangan prima p dan q harus tetap dirahasiakan, karena dengan mengetahui a, p dan q maka d dapat dicari dari persamaan

$$ad \equiv 1 \pmod{(p-1)(q-1)},$$

dan M_i dari persamaan

$$(M_i^{\,a})^d \equiv M_i \text{ (modulo } n)$$

sehingga pesan yang semula dirahasiakan akan dapat dibaca oleh ahli cryptanalysis.

Mungkinkah ada cara lain bagi ahli cryptanalysis untuk membaca pesan sandi dari sistem penyandian RSA,

selain dengan menghitung p dan q ? Bilangan d dapat dihitung jika (p-1)(q-1) diketahui.

Bila (p-1)(q-1) diketahui, maka p dan q dapat dihitung dengan langkah-langkah sebagai berikut :

$$(p-1)(q-1) = pq - (p+q) + 1$$
$$= n - (p+q) + 1$$

Jadi jika (p-1)(q-1) diketahui, maka (p + q) dapat dihitung.

Selanjutnya p - q dapat dihitung dari persamaan

$$(p - q)^2 = (p + q)^2 + 4pq$$
$$= (p + q)^2 + 4n$$

dan p dan q diperoleh dari

$$p = 1/2 [(p + q) + (p - q)]$$
$$q = 1/2 [(p + q) - (p - q)].$$

Jadi usaha untuk menemukan nilai (p-1)(q-1) adalah ekuivalen dengan usaha untuk menemukan kedua faktor prima p dan q dari n itu.

Sebenarnya seberapa sulitkah memfaktorkan suatu bilangan komposit dengan digit yang sangat besar ? Para ahli matematika seperti perancang sistem penyandian ini percaya bahwa masalah pemfaktoran ini pada hakekatnya sangat sulit dan bahwa belum ada sebuah algoritma yang benar-benar efisien untuk menemukan faktor-faktor

prima dari sebuah bilangan komposit. Berikut ini akan disajikan tentang hasil sebuah penelitian banyaknya langkah dan lamanya waktu yang diperlukan untuk menemukan faktor prima suatu bilangan komposit dengan digit yang besar.[6]

Banyaknya digit	Banyaknya langkah dalam algoritma	Lamanya waktu pengerjaan
50	$1,4 \times 10^{10}$	3,9 jam
75	$9,0 \times 10^{15}$	104 hari
100	$2,3 \times 10^{15}$	74,0 tahun
200	$1,2 \times 10^{23}$	$3,8 \times 10^{9}$ tahun
300	$1,5 \times 10^{29}$	$4,8 \times 10^{15}$ tahun
500	$1,3 \times 10^{39}$	$4,2 \times 10^{25}$ tahun

tabel 5.1

Rivest, perancang sistem penyandian ini, memperkirakan bahwa untuk menemukan faktor-faktor prima dari sebuah bilangan yang berdigit 125 atau 126, dengan algoritma yang terbaik dan komputer yang tercepat pada saat ini dibutuhkan waktu sekitar 40 quadrillion tahun (4.10^{16} tahun)! Hal itu berbeda sekali dengan kecepatan menemukan bilangan prima pertama setelah 2^{200} (yaitu $2^{200} + 235$ yang berdigit 61) yang hanya sekitar 45 detik

---

6 Alan G Konheim [1981], hal 303

saja. [7]

George Mackiw menggarisbawahi kenyataan tersebut :

Meskipun banyak algoritma untuk memfaktorkan suatu bilangan bulat n telah dikembangkan, tapi tidak ada yang kecepatannya mendekati kecepatan algoritma untuk memeriksa apakah suatu bilangan itu prima atau bukan.[8]

Dari uraian di atas dapat dikatakan bahwa dengan menggunakan Sistem Penyandian dari RSA ini kita dapat membuat pesan sandi yang praktis tidak dapat dipecahkan oleh pihak lawan.

---

7 Martin Gardner [1977], hal.123

8 George Mackiw [1985], hal 126

BAB VI

KESIMPULAN, IMPLIKASI DAN SARAN

A. Kesimpulan

Suatu sistem penyandian dapat dikatakan aman, jika ahli cryptanalysis pihak lawan dengan segala daya upayanya tidak dapat membaca pesan sandi yang dikirim dengan sistem tersebut. Salah satu sistem penyandian yang demikian itu adalah Sistem Penyandian dari Rivest, Shamir dan Adleman. Sistem penyandian ini menggunakan teori bilangan sebagai landasan teorinya, khususnya algoritma Euclides, bilangan-bilangan modulo, serta bilangan prima.

Dalam sistem penyandian RSA ini sipemakai hanya perlu merahasiakan dua buah bilangan prima p dan q saja. Dasar pemikiran dari sistem penyandian ini adalah sulitnya menemukan faktor prima dari suatu bilangan komposit dengan digit yang besar.

Sistem penyandian RSA, ini menggunakan bilangan-bilangan yang telah dikacaukan dengan memangkatkannya dengan suatu bilangan eksponen, sebagai pesan sandinya.

Kemungkinan sistem penyandian ini dapat dibaca oleh pihak lawan tergantung pada pemilihan dua buah bilangan primanya. Semakin besar digit bilangan prima tersebut tentu semakin sulit pihak lawan memecahkan pesan sandi tersebut; bahkan dengan besar digit tertentu pesan sandi tersebut praktis menjadi tidak terpecahkan lagi.

Sistem penyandian ini sangat istimewa, karena sipemakai hanya perlu merahasiakan dua buah bilangan prima saja, sedangkan unsur lainnya di dalam sistem ini boleh diketahui oleh pihak lawan.

B. Implikasi

Sistem penyandian RSA ini dapat digunakan jika kita ingin merahasiakan suatu pesan yang sangat penting dalam suatu komunikasi. Dirancangnya sistem penyandian ini, mendorong para ahli cryptanalysis untuk terus berusaha merancang suatu algoritma yang efisien dan efektif untuk menemukan faktor-faktor prima dari suatu bilangan komposit dengan digit besar.

C. SARAN

Dari uraian di atas, diajukan saran-saran berikut :

- agar sistem ini dapat terus dikembangkan sesuai dengan kemajuan ilmu dan teknologi;
- agar para sarjana matematika termotivasi untuk terus mengembangkan terapan matematika di berbagai bidang, walaupun matematika sendiri merupakan ilmu yang abstrak.

LAMPIRAN A

TABEL ASCII

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	0		27	1B	←	54	36	6	81	51	Q
1	1	☺	28	1C	└	55	37	7	82	52	R
2	2	☻	29	1D	↔	56	38	8	83	53	S
3	3	♥	30	1E	▲	57	39	9	84	54	T
4	4	♦	31	1F	▼	58	3A	:	85	55	U
5	5	♣	32	20		59	3B	;	86	56	V
6	6	♠	33	21	!	60	3C	<	87	57	W
7	7	•	34	22	"	61	3D	=	88	58	X
8	8	◘	35	23	#	62	3E	>	89	59	Y
9	9	○	36	24	$	63	3F	?	90	5A	Z
10	A	◙	37	25	%	64	40	@	91	5B	[
11	B	♂	38	26	&	65	41	A	92	5C	\
12	C	♀	39	27	'	66	42	B	93	5D	]
13	D	♪	40	28	(	67	43	C	94	5E	^
14	E	♫	41	29	)	68	44	D	95	5F	_
15	F	☼	42	2A	*	69	45	E	96	60	`
16	10	►	43	2B	+	70	46	F	97	61	a
17	11	◄	44	2C	,	71	47	G	98	62	b
18	12	↕	45	2D	-	72	48	H	99	63	c
19	13	‼	46	2E	.	73	49	I	100	64	d
20	14	¶	47	2F	/	74	4A	J	101	65	e
21	15	§	48	30	0	75	4B	K	102	66	f
22	16	▬	49	31	1	76	4C	L	103	67	g
23	17	↨	50	32	2	77	4D	M	104	68	h
24	18	↑	51	33	3	78	4E	N	105	69	i
25	19	↓	52	34	4	79	4F	O	106	6A	j
26	1A	→	53	35	5	80	50	P	107	6B	k

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
108	6C	l	145	91	æ	182	B6	╢	219	DB	▓
109	6D	m	146	92	Æ	183	B7	╖	220	DC	▄
110	6E	n	147	93	ô	184	B8	╕	221	DD	▌
111	6F	o	148	94	ö	185	B9	╣	222	DE	▐
112	70	p	149	95	ò	186	BA	║	223	DF	▀
113	71	q	150	96	û	187	BB	╗	224	E0	α
114	72	r	151	97	ù	188	BC	╝	225	E1	ß
115	73	s	152	98	ÿ	189	BD	╜	226	E2	Γ
116	74	t	153	99	Ö	190	BE	╛	227	E3	π
117	75	u	154	9A	Ü	191	BF	┐	228	E4	Σ
118	76	v	155	9B	¢	192	C0	└	229	E5	σ
119	77	w	156	9C	£	193	C1	┴	230	E6	µ
120	78	x	157	9D	¥	194	C2	┬	231	E7	τ
121	79	y	158	9E	₧	195	C3	├	232	E8	Φ
122	7A	z	159	9F	ƒ	196	C4	─	233	E9	θ
123	7B	{	160	A0	á	197	C5	┼	234	EA	Ω
124	7C	|	161	A1	í	198	C6	╞	235	EB	δ
125	7D	}	162	A2	ó	199	C7	╟	236	EC	∞
126	7E	~	163	A3	ú	200	C8	╚	237	ED	ø
127	7F	⌂	164	A4	ñ	201	C9	╔	238	EE	∈
128	80	Ç	165	A5	Ñ	202	CA	╩	239	EF	∩
129	81	ü	166	A6	ª	203	CB	╦	240	F0	≡
130	82	é	167	A7	º	204	CC	╠	241	F1	±
131	83	â	168	A8	¿	205	CD	═	242	F2	≥
132	84	ä	169	A9	⌐	206	CE	╬	243	F3	≤
133	85	à	170	AA	¬	207	CF	╧	244	F4	⌠
134	86	å	171	AB	½	208	D0	╨	245	F5	⌡
135	87	ç	172	AC	¼	209	D1	╤	246	F6	÷
136	88	ê	173	AD	¡	210	D2	╥	247	F7	≈
137	89	ë	174	AE	«	211	D3	╙	248	F8	°
138	8A	è	175	AF	»	212	D4	╘	249	F9	∙
139	8B	ï	176	B0	░	213	D5	╒	250	FA	·
140	8C	î	177	B1	▒	214	D6	╓	251	FB	√
141	8D	ì	178	B2	▓	215	D7	╫	252	FC	ⁿ
142	8E	Ä	179	B3	│	216	D8	╪	253	FD	²
143	8F	Å	180	B4	┤	217	D9	┘	254	FE	■
144	90	É	181	B5	╡	218	DA	┌	255	FF	

LAMPIRAN B

PROGRAM BASIC UNTUK MENGHITUNG R^A (MOD N)

```
10  CLS
20  REM PROGRAM BASIC UNTUK MENGHITUNG R^A (MOD N)
30  INPUT " R = ";R : INPUT "A = ";A : INPUT "N = ";N
40  J = 1
50  Q = INT(A/2)  : S = A - Q * 2
60  IF S = 1 THEN J = J * R
70  J = J - INT(J/N) * N
80  IF Q = 0 THEN 120
90  A = Q
100 R = R * R
110 GO TO 50
120 PRINT J : END
```

DAFTAR PUSTAKA

Albertson,Michael O dan Joan P. Hutchinson
1988 Discrete Mathematics With Algorithms. New York: John Wiley & Sons

Burn, R.P.
1982 A Pathway Into Number Theory. Cambridge: Cambridge University Press

Gardner, Martin
1977 "Mathematical Games: A new kind of cipher that would take millions of years to break" dalam: Scientific American. Vol.237 No.2 : 120-123

Hellman, Martin E
1979 "The Mathematics of Public-Key Cryptography" dalam: Scientific American. Vol.241 No. 2 : 130-139

Konheim, Alan G.
1981 Cryptography A Primer. New York: John Wiley & Sons

Mackiw, George.
1985 Applications of Abstract Algebra. New York: John Wiley & Sons

Manber, Udi
1989 Introduction to Algorithms A Creative Approach. Reading: Addison Wesley

NN
1984 "Cracking a Record Number" dalam : Time Vol. 123 No.7 : 41

Pomerance, Carl
1982 "The Search for Prime Numbers" dalam: Scientific American. Vol. 247 No. 6 :122-130

Sedgewick, Robert
1990 Algorithms in C. Reading : Addison Wesley

Smith, Jefrey D
1989 Design and Analysis of Algorithms.
Boston : PWS Kent

Swain, Robert L.
1963 Understanding Arithmetic. New York :
Holt, Rinehart and Winston