PENERAPAN METODA CHINESE REMAINDER THEOREM PADA RSA

Yuri Andri Gani – 13506118

Sekolah Teknik Elektro dan Informatika ITB, Bandung, 40132, email: if16118@students.if.itb.ac.id

Abstrak

Algoritma RSA merupakan salah satu jenis algoritma dalam sistem kriptografi kunci-publik, atau dikenal juga dengan kriptografi asimetris yang adalah bentuk kriptografi dimana pengguna memiliki pasangan kunci kriptografi yaitu kunci publik dan kunci privat. Kunci privat dirahasiakan, sedangkan kunci publik dapat disebarluaskan. Sebuah pesan yang dienkripsi dengan kunci publik hanya dapat didekripsi dengan kunci privat yang berkoresponden. Algoritma RSA merupakan algoritma pertama yang diketahui cocok untuk digital signature maupun untuk enkripsi, dan merupakan salah satu dari kemajuan besar dari sistem kriptografi kunci publik. Algoritma RSA secara luas digunakan pada protokol electronic commerce,. dan dipercaya aman jika diberikan kunci yang panjang dan penggunaan implementasi yang up-to date. Pada abad pertama, seorang matematikawan China

yang bernama Sun Tse mengajukan pertanyaan sebagai berikut yang akan dikenal sebagai Chinese Remainder Problem (CRT) : " Tentukan sebuah bilangan bulat yang bila dibagi dengan 5 menyisakan 3, bila dibagi 7 menyisakan 5, dan bila dibagi 11 menyisakan 7 "; dan mempelopori Chinese Remainder Theorem yang dapat digunakan untuk menyelesaikan masalah diatas yaitu : Misalkan m1, m2, ...,mn adalah bilangan bulat positif sedemikian sehingga FPB(mi, mj)=1 untuk i ≠ j. Maka sistem kongruen lanjar x ≡ ak (mod mk) mempunyai sebuah solusi unik modulo m=m1 · m2 · ...·mn

Dalam karya tulis ini akan dibahas tentang algoritma RSA standar antara lain proses enkripsi dan dekripsinya. Kemudian akan dibahas Chinese Remainder Theorem. Setelah itu akan dibahas bagaimana implementasi Chinese Remainder Theorem pada algoritma RSA antara lain dalam pembangkitan kuncinya.

RSA

Algoritma RSA merupakan singkatan dari nama belakang para penemunya, yaitu antara lain Ron Rivest, Adi Shamir, dan Len Adleman. Algoritma RSA ini merupakan algoritma kriptografi kunci publik yang populer digunakan untuk otentikasi keaslian suatu data digital. Algoritma RSA termasuk bagian dari web browser dari Microsoft dan Netscape dan digunakan oleh SSL (Secure Socket Layer) yang menjamin keamanan dan privasi di internet. Metode ini didasarkan pada ide bahwa mengalikan dua bilangan dapat mudah dilakukan, khususnya dengan perangkat komputer. Tetapi memfaktorkan bilangan dapat jadi sulit dilakukan". Contohnya, mudah dilakukan untuk mengambil dua bilangan prima misalnya x dan y dan menghitung hasil operasi kalinya N =xy. Tetapi jika diberikan nilai N, akan sulit untuk menemukan faktor-faktornya yaitu x dan y, terutama untuk bilangan N yang besar. Enkripsi menggunakan nilai atau kunci publik (public key) yang disebarluaskan dan diketahui semua orang yang ingin mengirim pesan. Sedangkan dekripsinya menggunakan sebuah kunci pribadi (private key) yang dijaga kerahasiannya oleh penerima dan tidak dapat dideduksi dari kunci publik. Sistem kriptografi dengan kunci publik seperti halnya algoritma RSA ini bekerja tanpa mengharuskan kedua pihak menjaga kerahasiaan, kunci pribadi tidak perlu diberitahu ke pengirim pesan. Keamanan enkripsi dan dekripsi data dari algoritma RSA terletak pada kesulitan untuk memfaktorkan modulus N yang sangat besar. Besarnya bilangan yang digunakan mengakibatkan lambatnya operasi yang melibatkan algoritma RSA ini. Berikut skema algoritma RSA

Algoritma Pembangkitan Kunci

1. Ambil dua buah bilangan prima sembarang, p dan q
2. Hitung n = pq dan m = (p-1)*(q-1).
3. Pilih bilangan integer e, 1 < e < m, yang relatif prima terhadap m yaitu FPB(e,m) = 1.
4. Hitung eksponen rahasia d, 1 < d < m, sehingga ed ≡1 (mod m).

5. Kunci publik adalah (n,e) dan kunci privat adalah (n,d). Nilai p, q dan m juga harus dirahasiakan.
- n disebut juga modulus
- e disebut juga public exponent atau exponent enkripsi
- d disebut juga secret exponent atau exponent dekripsi

Enkripsi

1. Ambil kunci public (n,e).
2. Nyatakan plainteks dalam integer positif m
3. Enkripsi menjadi cipher teks c = me mod n

Dekripsi

1. Menggunakan kunci privat (n, d) untuk dekripsi dengan rumus m = cd mod n.

Contoh

Berikut contoh pemakaiannya :
Misalkan p dan q bilangan prima, p = 47 dan q =71 maka dapat dihitung

n = p q = 3337 dan
m = (p – 1) (q – 1) = 3220.

Pilih kunci publik e = 79 (yang relatif prima dengan 3220). Nilai e dan m dapat dipublikasikan ke umum. Selanjutnya akan dihitung kunci dekripsi d

e d ≡ 1 (mod m)

Kunci dekripsi d sebagai berikut:

d = (1 + (k m)) / e
d = (1 + (k 3220)) / 79

Dengan mencoba nilai-nilai k = 1, 2, 3, …, diperoleh nilai d yang bulat adalah 1019. Ini adalah kunci dekripsi.
Misalkan terdapat plainteks yang sudah dikonversi ke ASCII :

P = 7265827332737873

Pecah P menjadi blok yang lebih kecil (3 digit):

p1 = 726 p4 = 273
p2 = 582 p5 = 787
p3 = 733 p6 = 003

Blok pertama dienkripsikan sebagai 72679 mod 3337 = 215 = c1. Blok kedua dienkripsikan sebagai 58279 mod 3337 = 776 = c2. Dengan melakukan proses yang sama untuk sisa blok lainnya, dihasilkan chiperteks

C = 215 776 1743 933 1731 158.

Proses dekripsi dilakukan dengan menggunakan kunci rahasia d = 1019. Blok c1 didekripsikan sebagai 2151019 mod 3337 = 726 = p1, Blok c2 didekripsikan sebagai 7761019 mod 3337 = 582 = p2. Blok plainteks yang lain dikembalikan dengan cara yang serupa. Akhirnya kita memperoleh kembali plainteks semula

P = 7265827332737873

Aplikasi RSA digunakan antara lain pada :
- *Electronic mail*
- *Electronic data transfer*
- *Electronic data interchange*
- Distribusi *software*
- *Data storage*
- Aplikasi yang membutuhkan jaminan integritas data dan otentikasi data asli
- *Digital signature*

CHINESE REMAINDER THEOREM (CRT)

Sebuah permasalahan berikut dikemukakan oleh Sun Tse dalam bukunya Sunzi Suanjing :
" *Tentukan sebuah bilangan bulat yang bila dibagi dengan 5 menyisakan 3, bila dibagi 7 menyisakan 5, dan bila dibagi 11 menyisakan 7* ".

Persoalan jenis ini merupakan suatu contoh permasalahan yang secara luas dikenal sebagai Chinese Remainder Theorem.

Theorem 1 (Chinese Remainder Theorem)
Terdapat bilangan-bilangan $n_1,n_2,\ldots,n_k$ *adalah bilangan bulat positif di mana relatif prima pada pasangan. Contohnya :*$gcd(n_i,n_j) = 1$ *di mana* $i \neq j$. *Lebih jauh lagi,* $n = n_1 n_2 \ldots n_k$ *dan* $x_1, x_2; \ldots, x_k$ *adalah bilangan bulat. Maka sistem kongruen*

$$\begin{aligned} x &\equiv x_1 \bmod n_1 \\ x &\equiv x_2 \bmod n_2 \\ &\vdots \\ x &\equiv x_k \bmod n_k \end{aligned}$$

memiliki solusi yang simultan pada semua kongruen dan dua solusi apapun adalah saling kongruen

modulo. Lebih jauh lagi terdapat tepatnya satu solusi antara 0 dan n-1.

Solusi unik dari kongruen simultan memenuhi
0 <= x <= n dapat dihitung dengan :

$$x = \left(\sum_{i=1}^{k} x_i r_i s_i \right) \bmod n$$
$$= (x_1 r_1 s_1 + x_2 r_2 s_2 + \ldots + x_k r_k s_k) \bmod n$$

persamaan 1

Dimana $r_i = n/n_i$ dan $s_i = r_i^{-1} \bmod n_i$ untuk $i = 1,2,..,k$

Jika bilangan bulat n_1, n_2,…, n_k adalah pasangan relatif prima dan $n = n_1 n_2 \ldots n_k$, maka untuk semua bilangan bulat a, b pasti akan valid di mana a ≡ b mod n jika dan hanya jika a ≡ b mod n_i untuk setiap i = 1, 2,...,k..

Sebagai konsekuensi dari CRT, setiap bilangan bulat positif a < n dapat direpresentasikan secara unik sebagai sebuah k-tuple [a_1, a_2, ...,a_k] dan sebaliknya. Di mana a_i menunjukan sisa / residu a mod n_i untuk setiap i = 1, 2, …,k. Konversi a menjadi sistem residu didefinisikan dengan n_1, n_2, ..., n_k dilakukan secara sederhana dengan reduksi modular a mod n_i. Konversi balik dari representasi sisa menjadi "angka-angka standar" adalah lebih sulit seperti yang dibutuhkan dalam kalkulasi pada persamaan di atas.

Kelebihan Chinese Remainder Theorem adalah sebagai berikut :

- Mempercepat untuk operasi kunci pribadi (dekripsi, pemberian tandatangan digital).
- Dua n/2-bit eksponensials mod P dan mod Q. sebagai ganti satu n-bit eksponensial mod N (N=P*Q).
- Split n-bit multiplier hardware ke dalam dua n/2-bit pengali, melaksanakan n/2-bit eksponensials paralel.
- Kombinasi hasil menurut CRT.
- CRT meningkat/kan decryption melewati suatu faktor aproksimasi 3- 3.5.

Kalkulasi penting di dalam rencana enkripsi RSA adalah eksponensial modular M= E_d(Mod n). Ini dilakukan setiap kali bagian dari pesan dilakukan enkripsi/dekripsi. d dan n adalah bilangan bulat yang sangat besar, oleh karena itu operasi ini sangat mahal. Sehingga harus ditemukan alternatif metoda biner untuk eksponensial modular.

Keuntungan dasar dengan menggunakan Chinese Remainder Theorem adalah memungkinkan untuk membagi modulo eksponensial yang besar ke dalam dua eksponensial yang jauh lebih kecil, satu di atas p dan satu di atas q. Dua modulo ini adalah faktor utama dari n yang dikenali. Kemudian masalah mengurangi ukuran dengan penggunaan teoreme Fermat's yang lebih kecil. Metoda ini pertama diusulkan oleh Quisquater dan Couvreur.

Jika kita menggunakan penyajian residu { r1,r2,…, rk} pada x, CRT memungkinkan untuk menentukan | x| yang disajikan faktor umum terbesar mengenai seluruh pembawa modulo 1 (yaitu (r_i, r_j)= 1, i ≠ j). Modulo dikenal sebagai operasi memasangkan bilangan prima secara relatif

Hal penting yang terdapat pada bagian ini adalah bahwa jika jumlah x dibagi ke dalam bentuk residunya, operasi pada residu itu dapat dilaksanakan bebas tiap satu dan lainnya. Sekali ketika menyelesaikan proses residu, jawaban akhir menggunakan CRT dapat direkonstruksi. Diketahui bahwa residu x adalah jauh lebih kecil dari x dirinya sendiri, oleh karena itu operasi individu akan memiliki kompleksitas yang jauh lebih kecil.
Membagi operasi modulo adalah dua kalkulasi mandiri. Sebagai ganti dilakukannya eksponensial (mod n), x dibagi ke dalam (mod p) dan (mod q). Karena kedua-duanya p dan q adalah utama, mereka mencukupi kebutuhan untuk merekonstruksi menggunakan CRT. Hal tersisa yang harus dilakukan kini tinggal melakukan eksponensial keduanya sebagai berikut:

$$M_p = E_d \pmod{p}$$
$$M_q = E_d \pmod{q}$$

Kita kemudian bisa menerapkan CRT untuk merekonstruksi pesan akhir dari Mp dan Mq.

Algoritma dari CRT dengan x_1,…,x_k sebagai residu dan moduli n_1,…,n_k sebagai input dan mengomputasikan x, maka solusi dari sistemnya adalah :

INPUT : x_1,…,x_k, n_1,…,n_k;
OUTPUT : x memeriksa x = x_i mod m_i untuk i = 1 sampai k;
x <- 0
n <- n_1
for i =2 to k do
    n <- n * n_i
    for i = 1 to k do

N_i <- n/n_i;
y_i <- N_{i-1} mod n_i;
G <- y_i * M_i mod m;
G <- G * x_i mod m;
X <- x + G mod m;

Return (x);

Kasus khusus dari CRT bila modulus adalah hasil dari dua bilangan prima : N = PQ. Jika ingin dikomputasikan M menjadi seperti M = M_P mod P dan M = M_Q mod Q maka :

INPUT: M_P ,M_Q, P,Q,N;
OUTPUT: M;
Y <- Q_{-1} mod P;
M <- y * Q mod N;
M <- M * M_P mod N;
y <- P_{-1} mod Q;
G <- y * P mod N;
G <- G * M_Q mod N;
M <- M + G;
return(M);

Pada langkah 1 dan 4 telah dikomputasikan dua invers dari bilangan bulat n/2-bit dan empat perkalian dari bilangan bulat n-bit pada langkah 2, 3, 5 dan 6. Setiap invers sama dengan 20 perkalian dari bilangan bulat n/2-Bit. Sehingga akhirnya diperoleh kompleksitas dari $28n_2$ + $o(n_2)$ dan sebuah pemanfaatan memori dari 4n (parameter sistem) + 3n/2 (akumulator).

Jika terdapat persamaan a = b mod n, maka dikatakan *'a congruent terhadap b mod n'* bila selisih a-b dapat dibagi oleh n sehingga : n | (a -b) atau a – b = k.n.
Beberapa sifat dari *congruent* adalah :

- a = b mod n jika dan hanya jika a dan b menghasilkan sisa yang sama jika dibagi dengan n
- reflexive, jika a = a mod n
- symmetry, jika a = b mod n, maka b = a mod n
- transive, jika a = b mod n dan b = c mod n, maka a = c mod n
- jika a = a_1 mod n dan b = b_1 mod n, maka a + b = a_1 + b_1 (mod n)

Kelas ekivalen dari bilangan bulat adalah semua bilangan bulat yang *congruent* ke a modulo n. Sedangkan bilangan bulat modulo n, dinyatakan Zn, adalah set dari kelas bilangan bulat ekivalen. Penjumlahan, pengurangan dan perkalian berlaku pada himpunan ini.

Contoh :Z_{25} ={0,1,2,....,24}
Pada operasi modular eksponensial, M=C_d mod n intinya adalah perulangan *modular multiflication*, nilai M tidak akan lebih dari n - 1.

IMPLEMENTASI CRT PADA RSA

Biasanya kunci publik e dari RSA adalah nilai yang relatif rendah, contohnya 216 + 1 (sebuah nilai yang standar). Sehingga, pada proses chiper (bukan pada proses pemberian tanda tangan digital) tidak akan diperoleh masalah dengan kecepatan chiper karena bilangan pangkat e akan relatif kecil.

Pada saat bilangan n = p*q menjadi jauh lebih besar, dengan urutan 21.024 jika membicarakan tentang kunci dengan panjang 1.024 bit, kunci privat d biasanya akan jauh lebih besar daripada nilai e dan nilai ini akan jatuh sangat dekat dengan nilai dari 1.024 bit. Oleh karena itu, akan sangat mahal bagi penerima pesan untuk mendekripsi pesan dengan kunci privat yang dimilikinya atau untuk menandatangani suatu dokumen dengan kunci privat tertentu.

Solusi yang ditawarkan adalah dengan menggunakan Chinese Remainder Theorem (CRT): daripada bekerja dengan nilai n, akan lebih baik bekerja dengan nilai p dan q sehingga perpangkatan modular akan dapat dilakukan dengan p dan q, jauh lebih cepat daripada menggunakan n.

Single Radix Conversion (SRC)
Implementasi CRT untuk mempercepat kriptografi RSA tidak hanya dilakukan dengan membagi kode pesan menjadi dua bagiantetapi dapat juga menggunakan satu langkah konversi dinamakan SRC. Berikut adalah langkah-langkah untuk SRC berlaku pula bagi RSA.

Langkah 1: membagi eksponensial kedalam bentuk (mod p) dan (mod q).
$M_p = E_d (\text{mod } p)$
$M_q = E_d (\text{mod } q)$

Langkah 2: kurangi kompleksitas dengan menerapkan Teorema Fermat's kepada eksponen, yang harus dihitung hanyalah:
$M_p = E_d^1 (\text{mod p})$
$M_q = E_d^2 (\text{mod } q)$
Dimana :
$d_1 = d \text{ mod } (p - 1)$
$d_2 = d \text{ mod}(q - 1)$

Ukuran d1, dan d2 kini separuh d. Karena kompleksitas tumbuh bersifat exponen terhadap d, hal ini akan menghasilkan tabungan data yang sangat besar.
Langkah 3: gunakan CRT untuk merekonstruksi pesan kita.
$M = M_p (q^{-1} \bmod p)q + M_q (p^{-1} \bmod q)p \pmod n$

Mixed Radix Conversion (MRC)
Decryption menggunakan MRC serupa dengan SRC . Satu-Satunya perbedaan untuk aplikasi ini adalah di dalam perhitungan rekonstruksi akhir.

Langkah 1:
Bagi eksponensial ke dalam (mod p) dan (mod q). Juga menerapkan Teorema Fermat's untuk mengurangi eksponen.
$M_p = E_{d^1} \pmod p$
$M_q = E_{d^2} \pmod q$

Langkah 2:
Rekonstruksi pesan menggunakan cara sebagai berikut :
$M = M_p + [(M_q - M_p) \times (p^{-1} \bmod q) \bmod q] \times p$
atau lebih umum menggunakan :
$M = M_q + [(M_p - M_q) \times (q^{-1} \bmod p) \bmod p] \times q$

Dekripsi RSA menggunakan CRT
Untuk mempercepat operasi dekripsi digunakan *Metoda Chinese Remainder Theorem*. Metode ini menggunakan faktor prima *mod n*, yaitu p dan q karena $n = p.q$, yang menyatakan sebagai berikut :
Agar $m_1, m_2,...,m_r$ relatif prima, maka sistem kongruen

$X = a_1 \bmod m_1$
$X = a_2 \bmod m_2$
$X = a_3 \bmod m_3$

mempunyai solusi unik untuk modulo $M = m_1, m_2, ..., m_r$.
Dikarenakan p dan q adalah bilangan prima, pesan apapun akan direpresentasikan secara unik dengan tuple $[M_P;M_Q]$, di mana $M_P = M$
mod P dan $M_Q = M \bmod Q$. Oleh karena itu, dimungkinkan juga untuk memperoleh nilai tersebut dengan komputasi $M_P;M_Q$ dan mengkombinasikan ulang sesuai dengan persamaan 1 di atas, daripada komputasi biasa menggunakan $M = C_D \bmod N$.

$$\begin{aligned} M_P &= M \bmod P = \left(C^D \bmod N\right) \bmod P \\ &= C^D \bmod P \ \ (\text{since } N = PQ) \\ &= C^{D \bmod (P-1)} \bmod P \\ &= C^{D_P} \bmod P \text{ with } D_P = D \bmod (P-1) \end{aligned}$$

persamaan 2

Lebih jauh lagi, akan mudah untuk diobservasi bahwa modulo P chiperteks C dapat dikurangi sebelum melakukan komputasi M_P, sehingga panjang dari semua operan dapat dikurangi menjadi setengahnya. Dengan persamaan $C_P = C \bmod P$ dan $C_Q = C \bmod Q$, sama seperti $D_P = D \bmod (P - 1)$ dan $D_Q = D \bmod (Q - 1)$, sehingga didapat persamaan di bawah ini untuk M_P dan M_Q :

$$M_P = {C_P}^{D_P} \bmod P \text{ and } M_Q = {C_Q}^{D_Q} \bmod Q$$

persamaan 3

Kombinasi dari M_P dan M_Q untuk mendapatkan M dapat dilakukan dengan menggunakan persamaan 1. Untuk kasus khusus untuk k = 2, $n_1 = P$, $n_2 = Q$ dan n = N = PQ, didapatkan $r_1 = N/P = Q$ dan $r_2 = N/Q = P$. Lebih jauh lagi, persamaan 1 dapat disederhanakan menggunakan teorema Fermat menjadi:

$$\begin{aligned} M &= \left(M_P Q \left(Q^{-1} \bmod P\right) + M_Q P \left(P^{-1} \bmod Q\right)\right) \bmod N \\ &= \left(M_P Q \left(Q^{P-2} \bmod P\right) + M_Q P \left(P^{Q-2} \bmod Q\right)\right) \bmod N \\ &= \left(M_P \left(Q^{P-1} \bmod N\right) + M_Q \left(P^{Q-1} \bmod N\right)\right) \bmod N \ (9) \end{aligned}$$

persamaan 4

Persamaan di atas muncul dari fakta bahwa a (b mod c) = (ab) mod (ac) untuk semua bilangan bulat positif a, b, c. Catatan : koefisien $Q_{P-1} \bmod N$ dan $P_{Q-1} \bmod N$ adalah konstan dan dapat di prekomputasi, sehingga usaha mengkombinasikan M_P dan M_Q dikurangi menjadi dua perkalian, satu sebagai penambahan dan satu sebagai pengurangan modulo N.

Ketika mengasumsikan bahwa eksponen $D_P = D \bmod (P - 1)$ dan $D_Q = D \bmod (Q - 1)$, sama seperti konstanta di mana diperlukan untuk rekombinasi $R_P = Q_{P-1} \bmod N$ dan $R_Q = P_{Q-1} \bmod N$ sudah diprekomputasikan, basis CRT untuk dekripsi RSA dapat dilakukan sesuai dengan langkah-langkah berikut:

1. Hitung $C_P = C \bmod P$ dan $C_Q = C \bmod Q$.
2. Hitung nilai eksponen $M_P = C_P$
3. Hitung koefisien $S_P = M_P R_P \bmod N$ dan $S_Q = M_Q R_Q \bmod N$.
4. Hitung $M = s_P + sQ$. Jika M >= N maka hitung

5. M = M - N.

Sekarang dapat dilihat secara jelas bahwa reduksi awal (langkah 1) dan kombinasi (langkah 3 dan 4) tidak mengakibatkan perubahan usaha komputasi yang signifikan bila dibandingkan dengan perhitungan pada langkah kedua. Dua eksponensial (langkah 2) dapat dikomputasikan secara independen satu sama lain dan secara paralel. Dibandingkan dengan dekripsi non-CRT pada n bit hardware, setiap eksponen CRT 4 kali lebih cepat jika n/2 bit hardware digunakan. Peningkatan kecepatan yang dramatis ini disebabkan pengurangan panjang 50 %, baik dari eksponen dan dari modulus. Melakukan dua eksponen seperti langkah 2 secara paralel membutuhkan dua n/2 bit pengali modular, menghasilkan sebuah faktor pemercepat 4 - ε, dengan ε dihitung untuk langkah 1, 2, dan 4.

KESIMPULAN

Dari hasil uraian uraian diatas dapat diambil kesimpulan antara lain :

1. Algoritma RSA adalah algoritma kriptografi kunci publik yang mengandalkan eksponensial modular.
2. Chinese Remainder Problem (CRT) dapat diaplikasikan untuk modifikasi algoritma salah satunya algoritma RSA.
3. Perbedaan algoritma RSA dengan CRT dengan algoritma RSA biasa terletak pada pembangkitan kunci dan proses dekripsi.
4. Algoritma RSA dengan CRT memiliki keuntungan dalam kecepatan proses bila dibandingkan dengan algoritma RSA standar.

DAFTAR PUSTAKA

[1]. Munir, Rinaldi, *Diktat Kuliah IF5054 Kriptografi*, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2006.

[2]. Munir, Rinaldi, *Diktat Kuliah IF2151 Matematika Diskrit*, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2005.

[3]. Iftene, Sorin, *Compartmented Secret Sharing Based on the Chinese Remainder Theorem,* Faculty of Computer Science ”Al. I. Cuza” University Iasi, Romania, 2005