Aplikasi Merkle-Hellman Knapsack Untuk Kriptografi File Teks

Akik Hidayat[1], Rudi Rosyadi [2], Erick Paulus[3]
Prodi Teknik Informatika, Fakultas MIPA, Universitas Padjadjaran
Jl. Raya Bandung Sumedang KM 21 Jatinangor Sumedang 45363
Email: [1]akik.hidayat@ymail.com, [2] rudirosadi@gmail.com, [3]erick.@unpad.ac.id

Abstrak – *Merkle-Hellman Knapsack merupakan kriptosistem yang menggunakan algoritma asimetris. Kelebihan algoritma asimetris ini adalah proses pendistribusian kunci pada media yang tidak aman seperti internet, tidak memerlukan kerahasiaan. Karena kunci yang didistribusikan adalah kunci publik. Sehingga jika kunci ini sampai hilang atau diketahui oleh orang lain yang tidak berhak, maka pesan sandi yang dikirim akan tetap aman. Sedangkan kunci private (rahasia) tetap disimpan (tidak didistribusikan). Dengan menggunakan Merkle-Hellman Knapsack dapat menggunakan ukuran kunci yang lebih kecil dibandingkan dengan kriptosistem seperti RSA. Kemampuan ini membuat Merkle-Hellman Knapsack mempunyai keamanan yang kuat dengan panjang kunci yang pendek.. Sedangkan tujuan yang ingin dicapai yaitu proses dekripsi dan enkripsi, mengaplikasikan metode kriptosistem Merkle-Hellman Knapsack menggunakan bahasa pemrograman C++.*

Kata kunci *: asimetris, enkripsi, dekripsi, kriptosistem, kunci publik, kunci private.*

1. Pendahuluan

Enkripsi adalah metode menguubah data pesan (*plaintext*) menjadi data sandi (*chipertext*), sedangkan dekripsi adalah metode merubah *chipertext* menjadi *plaintext*. Algoritma yang digunakan ada 2 (dua) macam yaitu algoritma simetris dan algoritma asimetris. Algoritma simetris adalah algoritma yang menggunakan kunci yang sama pada proses enkripsi dan dekripsinya. Sedangkan algoritma asimetris adalah algoritma yang menggunakan kunci publik pada proses enkripsi dan kunci private pada proses dekripsinya. Merkle-Hellman Knapsack merupakan kriptosistem yang menggunakan algoritma asymmetries . Kelebihan algoritma asimetris ini adalah proses pendistribusian kunci pada media yang tidak aman seperti internet, tidak memerlukan kerahasian. Karena kunci yang didistribusikan adalah kunci publik. Sehingga jika kunci ini sampai hilang atau diketahui oleh orang lain yang tidak berhak, maka pesan sandi yang dikirim akan tetap aman. Sedangkan kunci private tetap disimpan (tidak didistribusikan). Kelebihan lain adalah pada efisiensi jumlah kunci publik. Jika terdapat *n* user, maka hanya membutuhkan 1 (satu) kunci publik, sehingga untuk jumlah user yang sangat banyak, sistem ini sangat efisien. Menurut [5] menyatakan bahwa pada tingkat keamanan yang sama, Merkle-Hellman Knapsack dapat menggunakan ukuran kunci yang lebih kecil dibandingkan dengan kriptosistem seperti RSA. Kemampuan ini membuat Merkle-Hellman Knapsack mempunyai keamanan yang kuat dengan panjang kunci yang pendek [7]. Implementasi Merkle-Hellman Knapsack yang digunakan menggunakan logika xor. Panjang kunci yang digunakan antara 8 sampai 72 bit. Karena dalam Bahasa pemrograman Borland C++ tipe data yang paling tinggi adalah long double yang bisa menampung 18 digits. Misalnya saja dalam perhitungan perkalian antara 2 (dua) bilangan dengan panjang 9 digit akan menghasilkan bilangan dengan panjang 18 digit yang akan ditampung dalam tipe long double, kemudian dengan fungsi modulo akan dihasilkan kembali bilangan dengan panjang 9 digit.

2. Landasan Teori

2.1. Konsep Dasar Matematika Kriptografi

konsep dasar matematika yang berhubungan dengan persoalan kriptografi Merkle-Hellman Knapsack [6].

1. Aritmatika Modular
   Aritmatika Modular adalah operasi modulus yang merupakan sisa dari hasil pembagian bilangan bulat.
2. Bilangan prima
   Bilangan prima adalah bilangan bulat positif *p* (*p* > 1) yang pembaginya hanya 1 dan *p*.
3. Diskriminan
   Diskriminan adalah hasil kuadrat dari selisih akar-akar polinomial (suku banyak).

4. diberikan s1,s2,..,sn adalah himpunan bilangan positif (ukuran pemanggilan) dan T adalah bilangan positif, penyelesaian masalah adalah dengan mencari suatu vector 0-1 (x1,x2,..,xn) dapat dinyatakan sebagai berikut :
   $x_1s_1 + x_2s_2 + ... + x_ns_n = T.$ (2.1)
5. General Knapsack : yaitu diberikan $S = [a1,a2, \ldots, an]$, dan *T*, carilah vector *V* dari nilai 0 dan 1 seperti:

6. Superincreasing knapsacks

$$\sum_{i=1}^{n} a_i \; * \; vi = T \tag{2.2}$$

$$a_k > \sum_{j=1}^{k-1} a_j \tag{2.3}$$

$$s_k > \sum_{j=1}^{k-1} s_j \qquad \textbf{Super-Increasing Sets} \tag{2.4}$$

2.2. Algoritma Merkle-Hellman Knapsack

Apa yang kita butuhkan:

- S = (s1,…, Sn) bilangan integer superincreasing

$$p > \sum_{i=1}^{n} s_i \tag{2.5}$$

- bilangan prima

a, $1 \le a \le p\text{-}1$

- t = a si mod p

Public Key: t
Private Key: si, p, a, [1]

ISBN: 978-602-60581-0-2

Encode:

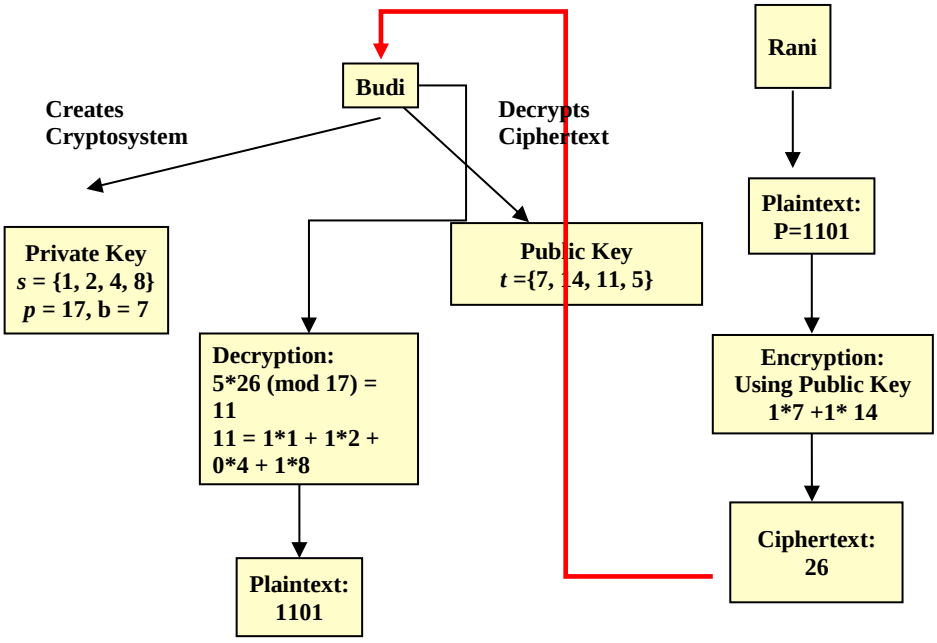
$e_s(x_1,\ldots,x_n) = \sum_{i=1}^{n} x_i t_i$

Decode:

$z = a^{-1} y \bmod p$

- penyelesaian masalah *subset* $(s_1,\ldots,s_n,Z)$ diberlakukan untuk $d_K(y) = (x_1,\ldots,x_n)$.[2]

2.3 Konsep Merkle-Hellman Knapsack



Gambar 1. Proses Plaintext menjadi ciphertext

3. Pembahasan

Mencari Knapsack superincreasing :

1. Pertama mengambil serangkaian ***superincreasing*** s bilangan bulat positif dengan cara pilih bilagan inisial(terkecil). Pilih bilangan selanjutnya dengan bilangan yang lebih besar daripada yang pertama. Kemudian pilih bilangan yang lebih besar daripada penjumlahan bilangan pertama dan kedua. Teruskan proses ini dari memilih bilangan-bilangan baru yang lebih besar daripada jumlah semua bilangan yang sebelumnya dipilih.

ISBN: 978-602-60581-0-2

2. Setelah memilih knapsack yang sederhana $S = [s_1, s_2,\ldots,s_m]$, kita memilih sebuah bilangan pengali *b* dan di modulus *p*.
   - Bilangan mod seharusnya adalah angka yang lebih besar daripada jumlah semua s_i dan merupakan bilangan prima
   - Pengali tidak mempunyai factor persekutuan dengan modulus.
3. Mencari kunci publik, mengganti setiap bilangan s_i dalam knapsack sederhana dengan ketentuan.
   $t_i = a * s_i \ mod \ p$
   S = [1, 2, 4, 8] dan diubah dari pengali *b* dan kemudian di mod *p* dimana b = 7 dan p = 17:
   1 * 7 = 7 mod 17 = 7
   2 * 7 = 14 mod 17 = 14
   4 * 7 = 28 mod 17 = 11
   9 * 7 = 63 mod 17 = 5
   knapsack t= [7,14,11,5]

4. Proses Enkripsi
   a. Pesan plaintext *P* bisa dituliskan dalam bentuk:
      $P = [p_1, p_2,..., p_k]$.
   b. Membagi pesan ke dalam blok bit-bit *m* , $P_0 = [p_l, p_2,..., p_m]$, $P_1 = [p_{m+1},..., p_{2m}]$, dan selanjutnya. *(m* adalah bilangan pembatas dalam knapsack)
   c. Memilih nilai dengan mengubah dari bentuk 1 bit kedalam P_i selanjutnya P_i disajikan sebagai vector yang dipilih untuk element t.
   d. Nilai ciphertext merupakan: *Pi* * t, target menggunakan blok P_i untuk memilih vector [4].
5. Proses Dekripsi :

Penerima tahu knapsack sederhana dan nilai dari ***a*** dan ***p*** yang ditranformasi ke dalam knapsack sulit.

- Dengan nilai a^{-1} kemudian $a^* a^{-1} = 1$ mod p. Dalam contoh kami, 7^{-1} mod 17 adalah 5, mulai 5 * 8 mod 17 = 40 mod 17 = (17 * 7) + 1 mod 17 = 1.
- Ingat bahwa *H* adalah knapsack sulit yang terjadi dari knapsack sederhana*S*. *H* adalah memperoleh *S* dengan

$H = w * S \ mod \ n$

- Pesan ciphertext di dapat dari algoritma enkripsi:
  $\boldsymbol{C = H * P = w * S * P \ mod \ n}$
- Untuk mengubah cipher, pengali *C* dari w^{-1}, mulai
  $\boldsymbol{w^{-1} * C = w^{-1} * H * P =}$
  $\boldsymbol{w^{-1} * w * S * P \ mod \ n =}$
  $\boldsymbol{S * P \ mod \ n}$

- Sekarang penerima dapat memecahkan masalah knapsack sederhana dengan knapsack *S* dan target $w^{-1} * C_i$ untuk beberapa bilangan ciphertext C_i.
- Dimulai $w^{-1} * C_i = S * P$ mod *n*, solusi untuk target $w^{-1} * C_i$ adalah blok plaintext P_i, yaitu adalah pesan asli yang di enkripsi.

4. Inplementasi Merkle-Hellman Knapsack

Diberikan Private key :

s = (1,2,5,11,32,87,141)
a = 200
p = 307

Plaintext (x) : SYARE

ISBN: 978-602-60581-0-2

Enkripsi Plaintext : SYARE
Enkripsi :
Perhitungan Public Key (t) :
ti = a* si mod p
$t_1 = a*s_1 \bmod p = 200 * 1 \bmod 307 = 200$
$t_2 = a*s_2 \bmod p = 200 * 2 \bmod 307 = 93$
$t_3 = a*s_3 \bmod p = 200 * 5 \bmod 307 = 79$
$t_4 = a*s_4 \bmod p = 200 * 11 \bmod 307 = 51$
$t_5 = a*s_5 \bmod p = 200 * 32 \bmod 307 = 260$
$t_6 = a*s_6 \bmod p = 200 * 87 \bmod 307 = 208$
$t_7 = a*s_7 \bmod p = 200 * 141 \bmod 307 = 263$
Didapatkan
t = (200,93,79,51,260,208,263)
Plaintext : SYARE (83 89 65 82 78)
Masing – masing kode ASCII tersebut dikonversi ke biner
S → 83 : 1010011
Y → 89 : 1011001
A → 65 : 1000001
R → 82 : 1010010
E → 69 : 1000101
Plaintext di bagi dalam block sesuai dengan banyaknya s,
pada contoh ini banyaknya s adalah 7 digit.
1010011→ y = 200 + 79 + 208 + 263 = 750
1011001→ y = 200 + 79 + 51 + 263 = 593
1000001→ y = 200 + 263 = 463
1010010→ y = 200 + 79 + 208 = 487
1000101→ y = 200 + 260 + 263 = 723
Didapatkan Ciphertext : 750 593 463 487 723

Dekripsi :
Hitung Z
$Z = a^{-1} y \bmod p$
200^{-1} =???? → dengan algoritma extended Euclidian
gcd (307,200) =
307 = 1 * 200 + 107
200 = 1 * 107 + 93
107 = 1 * 93 + 14
93 = 6 * 14 + 9
14 = 1 * 9 + 5
9 = 1 * 5 + 4
5 = 1 * 4 + 1
4 = 4 * 1 + 0
Selanjutnya:
$t_0 = 0$
$t_1 = 1$
$t_2 = t_0 – q_1 t_1 = 0 – 1*1 = -1$
$t_3 = t_1 – q_2 t_2 = 1 – 1*(-1) = 2$
$t_4 = t_2 – q_3 t_3 = (-1) – 1*2 = -3$
$t_5 = t_3 – q_4 t_4 = 2 – 6*(-3) = 20$
$t_6 = t_4 – q_5 t_5 = (-3) – 1*20 = -23$
$t_7 = t_5 – q_6 t_6 = 20 – 1*(-23) = 43$
$t_8 = t_6 – q_7 t_7 = (-23) – 1*43 = -66$

ISBN: 978-602-60581-0-2

$200^{-1} = 241$

- Untuk y = 750:
  Z = 241*750 mod 307 = 180750 mod 307= 234
  234 = 1*1 + 0*2 + 1*5 + 0*11 + 0*32 + 1*87 + 1*141

Plaintext → 1010011

- Untuk y = 593:
  Z = 241*593 mod 307 = 142913 mod 307 = 158
  158 = 1*1 + 0*2 + 1*5 + 1*11 + 0*32 + 0*87 + 1*141

Plaintext → 1011001

- Untuk y = 463:
  Z = 241*463 mod 30 = 111583 mod 307 = 142
  142 = 1*1 + 0*2 + 0*5 + 0*11 + 0*32 + 0*87 + 1*141

Plaintext → 1000001

- Untuk y = 487:
  Z = 241*487 mod 307 = 117367 mod 307 = 93
  93 = 1*1 + 0*2 + 1*5 + 0*11 + 0*32 + 1*87 + 0*141

Plaintext → 1010010

- Untuk y = 723:
  Z = 241*723 mod 307 = 174243 mod 307 = 174
  174 = 1*1 + 0*2 + 0*5 + 0*11 + 1*32 + 0*87 + 1*141

Plaintext → 1000101

Plaintext Dimasukkan dalam kode ASCII didapatkan (83 89 65 82 78)
Maka akan menjadi: SYARE

5. Kesimpulan

Merkle-Hellman Knapsack merupakan kriptosistem yang menggunakan algoritma asimetris. Kelebihan algoritma asimetris ini adalah proses pendistribusian kunci pada media yang tidak aman seperti internet, tidak memerlukan kerahasian. Karena kunci yang didistribusikan adalah kunci publik. Sehingga jika kunci ini sampai hilang atau diketahui oleh orang lain yang tidak berhak, maka pesan sandi yang dikirim akan tetap aman. Sedangkan kunci private tetap disimpan (tidak didistribusikan). Merkle-Hellman Knapsack punya Kelebihan lain pada efisiensi jumlah kunci publik. Jika terdapat *n* user, maka hanya membutuhkan 1 (satu) kunci publik, sehingga untuk jumlah user yang sangat banyak, sistem ini sangat efisien. Dengan adanya pertukaran kunci dalam enkripsi-dekripsi data dengan kriptosistem kurva elliptik adalah mengamanan data yang berupa teks untuk menghindari adanya penyadapan yang dilakukan oleh pihak-pihak yang tidak berkepentingan.

Daftar pustaka

[1]. A.Foster, (2004) *Apolynomial-time probabilistic algorithm for the mini-mum distance of anarbitrary linear error-correcting code*,Mathematics Honors Report,.

[2]. A.Meneezes, P. van Oorschot, and S.Vanstone, (1996) *Handbook of Applied Crytpography***.** CRC Press,.

[3]. C.A. van TILBORG,Henk, (2000). *Fundamentals of Cryptology*, Kluwer Academic Publishers,

[4]. KOBLITZ, Neal, (1987) *A Course in Number Theory & Cryptography*, Springer-Verlag New York Inc.,.

[5]. R.J.McEliece, (1978). *A public-key crypto system based on algebraic coding theory*, vol.42-44, DSN Progress Rep.,

[6]. Shala, Mahmet. (2004). *Primes Number In Cryptography*. University of Greenwich.

ISBN: 978-602-60581-0-2

[7]. Stinson, Douglas R.(1956). *Cryptography : Theory and Practice*. CRC Press. Boca Raton. Florida.

ISBN: 978-602-60581-0-2