PERTEMUAN 05

ALGORITMA KUNCI PUBLIK

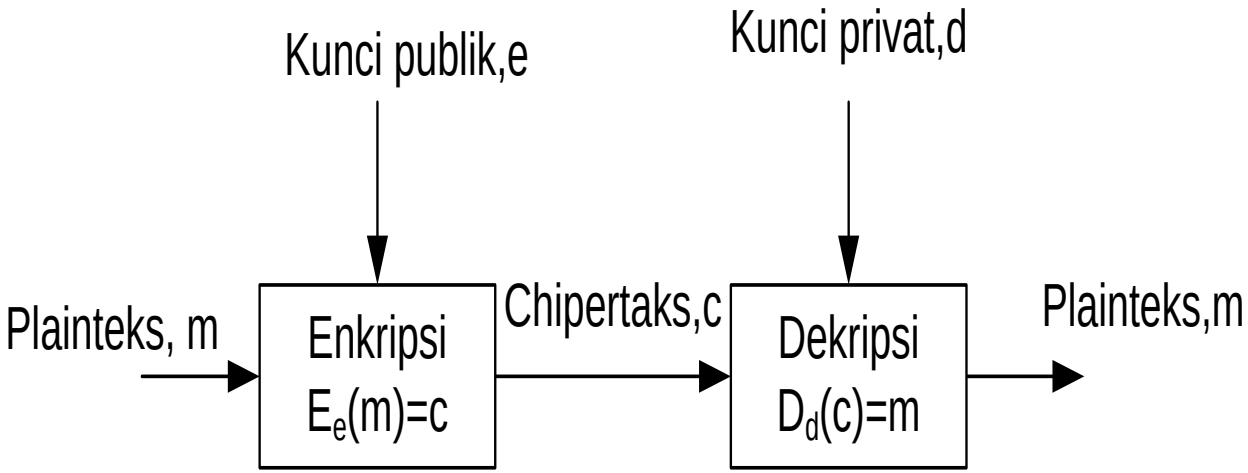
Sub Pembahasan

- ✓ Konsep kriptografi Kunci Publik
- ✓ Perbandingan Kriptografi Kunci Simetri dengan Kriptografi Kunci Publik
- ✓ Aplikasi Kriptografi Kunci Publik
- ✓ RSA
- ✓ ElGamal
- ✓ Algoritma Pertukaran Kunci Diffie-Hellman
- ✓ Algoritma Knapsack

- Kunci kriptografi dibuat sepasang, satu kunci untuk enkripsi, dan satu kunci untuk deskripsi
- Kunci untuk enkripsi disebut dengan kunci publik, disimbolkan dengan ***e***
- Kunci untuk dekripsi disebut dengan kunci privat, disimbolkan dengan ***d***
- Karena kunci enkripsi tidak sama dengan kunci deskripsi, maka sering disebut sistem kriptografi asimetri (Kunci publik)

Konsep Kriptografi Kunci Publik

- Konsep kriptografi kunci publik sederhana akan tetapi mempunyai konsekuensi pengunaan yang hebat.
- Misalkan E adalah kunci enkripsi, dan D adalah kunci deskripsi. E dan D adalah pasangan kunci untuk enkripsi dan deskripsi.



Sumber: V. Lusiana

Skema kriptografi kunci publik

- Konsep diatas digunakan untuk mengamankan pertukaran dari dua entitas yang saling berkomunikasi.
- Sistem kriptografi kunci publik cocok untuk kelompok pengguna dijaringan komputer (LAN, WAN)
- Sistem kunci publik tidak memerlukan pengiriman kunci privat melalui saluran komunikasi khusus
- Meskipun kunci publik di umumkan ke setiap orang didalam kelompok, namun perlu dilindungi agar otentikasinya terjamin. Misal tidak diubah oleh orang lain

Perbandingan Kriptografi Kunci Simetri dengan Kriptografi Kunci Publik (Asimetri)

- Kriptografi kunci simetri maupun asimetri mempunyai kelebihan dan kekurangan masing-masing

Kelebihan Kriptografi Kunci Simetri

- ✓ Algoritma dirancang sehingga proses enkripsi dan deskripsi membutuhkan waktu yang singkat
- ✓ Ukuran kunci relatif pendek
- ✓ Dapat digunakan untuk membangkitkan bilangan acak
- ✓ Dapat disusun untuk menghasilkan cipher yang lebih kuat
- ✓ Otentikasi pengirim pesan langsung diketahui dari cipherteks yang diterima, karena kunci hanya diketahui oleh pengirim dan penerima

Kelemahan kriptografi kunci simetri

- Harus dikirim melalui saluran yang aman, kedua pihak yang berkomunikasi harus menjaga kerahasiaan kunci
- Kunci harus sering diubah pada setiap komunikasi

Kelebihan Kriptografi Kunci Publik (Asimetri)

- ✓ Hanya kunci privat yang perlu dijaga kerahasiaannya.
- ✓ Pasangan kunci publik dan privat tidak perlu diubah bahkan dalam periode waktu yang panjang.
- ✓ Dapat digunakan untuk mengamankan pengiriman kunci simetri
- ✓ Beberapa algoritma dapat digunakan untuk pengiriman tanda tangan digital pada pesan.

Kelemahan kriptografi kunci Asimetri

- Enkripsi dan deskripsi data umumnya lebih lambat
- Ukuran cipherteks lebih besar daripada plainteks.
- Ukuran kunci lebih besar dari kunsi simetri
- Karena kunci diketahui secara luas, maka cipherteks tidak memberikan informasi mengenai otentikasi pengirim.

Kelemahan kriptografi kunci Asimetri

- Tidak ada algoritma kunci publik yang terbukti aman. Kebanyakan algoritma mendasarkan pada sulitnya memecahkan persoalan aritmetik yang menjadi dasar pembangkit kunci.

Aplikasi Kriptografi Kunci Publik

- Aplikasi kriptografi kunci publik dibagi menjadi tiga kategori:
  1. Enkripsi/Deskripsi
  2. Digital Signature
  3. Pertukaran Kunci (Key Exchange)
- Beberapa algoritma cocok digunakan untuk ketiga macam kategori aplikasi(contoh: RSA)
- Beberapa hanya ditujukan untuk aplikasi spesifik (contoh: DSA)

RSA

- ✓ Sandi RSA merupakan algoritma kriptografi kunci publik asimetri.
- ✓ Ditemukan pertama tahun 1977 oleh Ron Rivest, Adi Shamir dan Len Adleman.
- ✓ Nama RSA diambil dari nama tiga penemunya.

- ✓ Kunci enkripsi dan deskripsi menggunakan bilangan bulat.
- ✓ Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum, sehingga disebut kunci publik
- ✓ Sedangkan kunci deskripsi bersifat rahasia.
- ✓ Untuk menemukan kunci deskripsi, dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor primanya.
- ✓ Kekuatan algoritma ini terletak pada proses eksponensial, dan pemfaktoran bilangan menjadi 2 bilangan prima yang hingga kini perlu waktu yang lama untuk melakukan pemfaktorannya

- ✓ Skema RSA sendiri mengadopsi dari skema block cipher.
- ✓ Dimana sebelum dilakukan enkripsi, plainteks yang ada dibagi – bagi menjadi blok – blok dengan panjang yang sama.
- ✓ Plainteks dan cipherteksnya berupa integer(bilangan bulat) antara 1 hingga n.
- ✓ N berukuran biasanya sebesar 1024 bit, dan panjang bloknya sendiri berukuran lebih kecil atau sama dengan log(n) +1 dengan basis 2

Algoritma RSA

1. Menentukan dua bilangan prima
2. Menghitung nilai modulus
3. Menghitung nilai totient
4. Menentukan nilai ***e***
5. Mencari nilai ***d***
6. Mendapatkan nilai ***n,e, dan d*** sehingga pasangan kunci telah terbentuk.

ELGamaL

- Merupakan salah satu algoritma kriptografi kunci publik yang dibuat oleh Taher ElGamal pada tahun 1984.
- Algoritma ini pada umumnya digunakan untuk digital signature, tetapi kemudian dimodifikasi sehingga juga bisa digunakan untuk enkripsi dan deskripsi.
- Kekuatan algoritma ini terletak pada sulitnya menghitung logaritma diskrit.
- Algoritma ElGamal terdiri dari tiga proses, yaitu proses pembentukan kunci, enkripsi, dan deskripsi

1. Proses Pembentukan Kunci

- Algoritma ElGamal memerlukan sepasang kunci yang dibangkitkan dengan memilih sebuah bilangan prima ***p*** dan dua buah bilangan random ***g*** dan ***x***. Nilai ***g*** dan ***x*** lebih kecil dari ***p*** yang memenuhi persamaan :

$$y = g^x \bmod p$$

- Dari persamaan tersebut ***y***, ***g*** dan ***p*** merupakan kunci publik dan ***x*** adalah kunci rahasia

2. Proses Enkripsi

- Proses enkripsi merupakan proses mengubah pesan asli (plaintext) menjadi pesan rahasia (ciphertext).
- Pada proses ini digunakan kunci publik (***p, g, y*)**.

3. Proses Deskripsi

- Proses dekripsi merupakan proses mengubah pesan rahasia (ciphertext) menjadi pesan asli (plaintext).
- Pada proses ini digunakan kunci pribadi (***x, p***).

Algoritma Pertukaran Kunci Diffie-Hellman

- Berguna untuk berbagi kunci enkripsi simetri yang sama antara dua orang atau lebih.
- Keamanan algoritma ditentukan oleh sulitnya menghitung logaritma diskrit.

Parameter Umum

- Misalkan dua orang yang berkomunikasi: Alice dan Bob.
- Mula-mula Alice dan Bob menyepakati **bilangan prima** yang besar, *n* dan *g*, sedemikian sehingga $g < n$.
- Bilangan *n* dan *g* tidak perlu rahasia. Bahkan, Alice dan Bob dapat membicarakannya melalui saluran yang tidak aman sekalipun.

1. Alice membangkitan bilangan bulat acak yang besar x dan mengirim hasil perhitungan berikut kepada Bob:

   $X = g^x \bmod n$

2. Bob membangkitkan bilangan bulat acak yang besar y dan mengirim hasil perhitungan berikut kepada Alice:

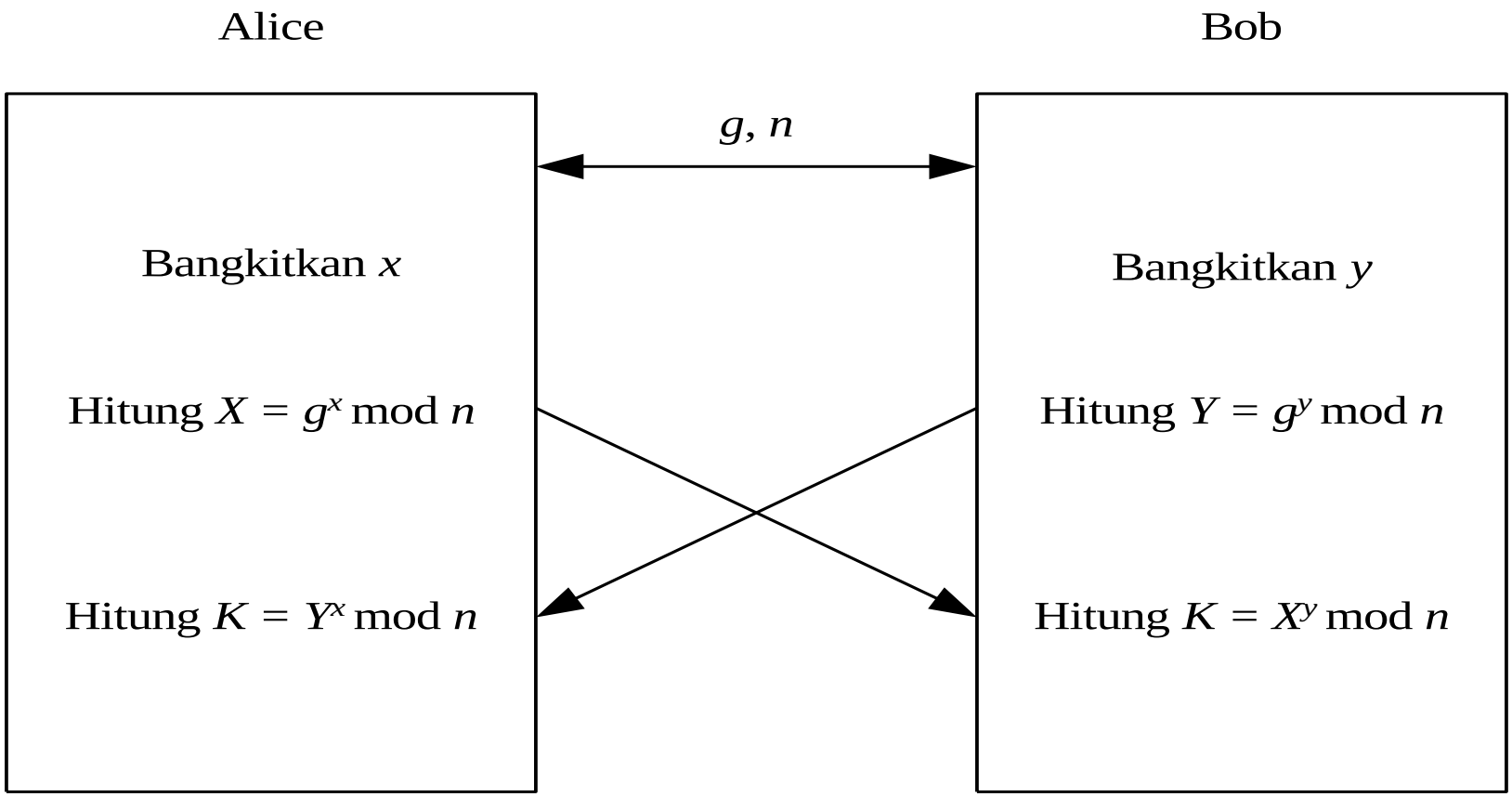
   $Y = g^y \bmod n$

3. Alice menghitung

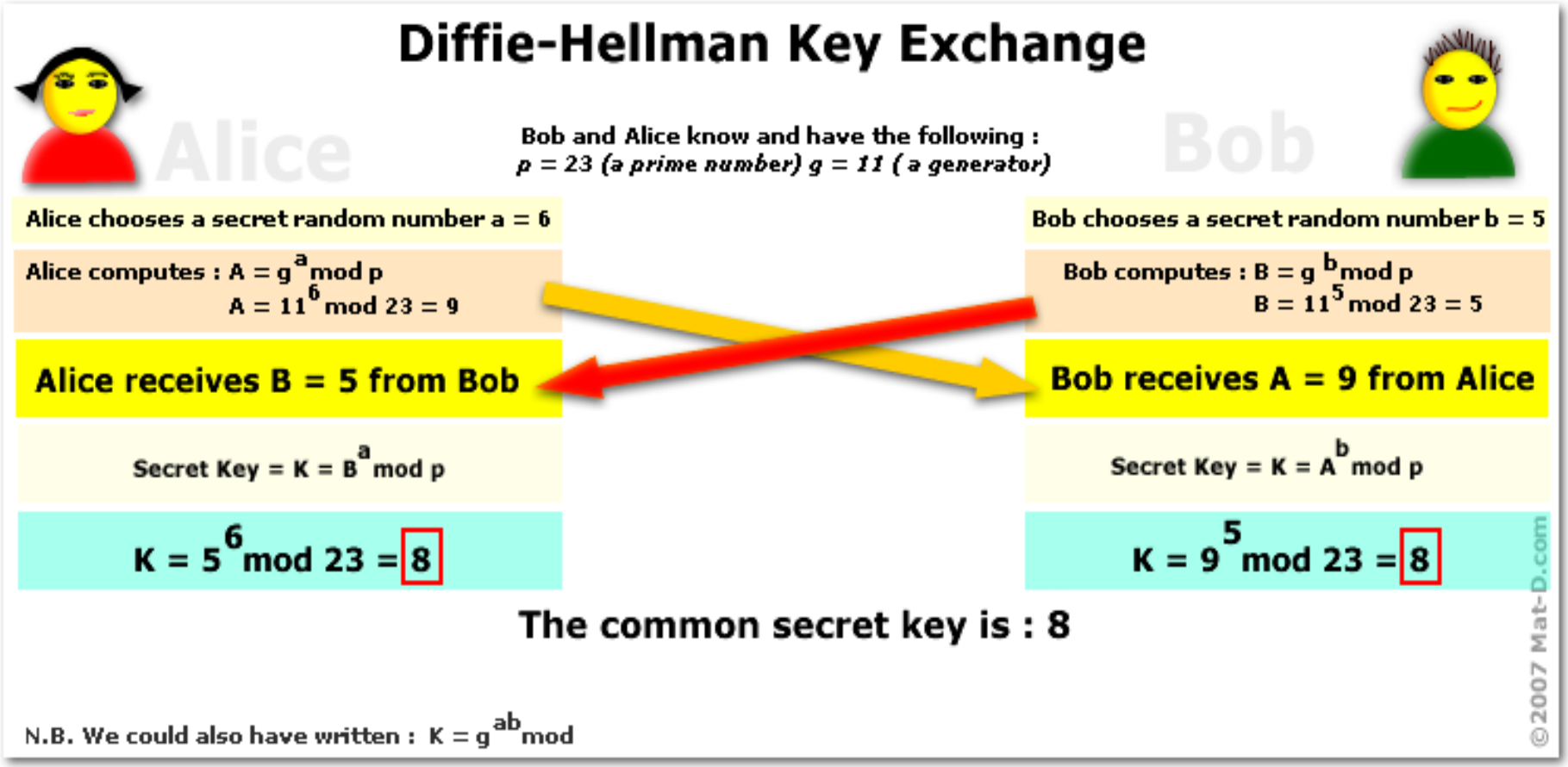
   $K = Y^x \bmod n$

4. Bob menghitung

   $K' = X^y \bmod n$



Contoh:



Sumber: http://www.mat-d.com/site/rsa-diffie-hellman-explained-in-3-minutes/

Algoritma Knapsack

- Knapsack dapat diartikan sebagai karung atau kantung.
- Karung digunakan untuk memuat sesuatu.
- Dan tentunya tidak semua objek dapat ditampung di dalam karung. Karung tersebut hanya dapat menyimpan beberapa objek dengan total ukurannya (weight) lebih kecil atau sama dengan ukuran kapasitas karung.
- Setiap objek itupun tidak harus kita masukkan seluruhnya. Tetapi bisa juga sebagian saja.

Knapsack Problem:

- ✓ Diberikan bobot knapsack adalah M.
- ✓ Diketahui *n* buah objek yang masing-masing bobotnya adalah w*1*, w*2*, ..., w*n*.
- ✓ Tentukan nilai b sedemikian sehingga M = b*1*w*1* + b*2*w*2*+ ... + b*n*w*n*.
- ✓ Yang dalam hal ini, b*i* bernilai 0 atau 1. Jika b*i*= 1, berarti objek *i* dimasukkan ke dalam knapsack, sebaliknya jika b*i* = 0, objek *i* tidak dimasukkan

- Ide dasar dari algoritma kriptografi knapsack adalah mengkodekan pesan sebagai rangkaian solusi dari dari persoalan knapsack. Setiap bobot widi dalam persoalan knapsack merupakan kunci privat, sedangkan bit-bit plainteks menyatakan b*i*.

- Misalkan n = 6.

  w1 = 1 w4 = 11

  w2 = 5 w5 = 14

  w3 = 6 w6 = 20.

- Plainteks: 111001010110000000011000
- Plainteks dibagi menjadi blok yang panjangnya n, kemudian setiap bit di dalam blok dikalikan dengan w*i* yang berkorepsonden sesuai dengan persamaan (1)

Blok plainteks ke-1: 111001

Knapsack : 1, 5, 6, 11, 14, 20

Kriptogram: $(1 \times 1) + (1 \times 5) + (1 \times 6) + (1 \times 20)$

$= 32$

Blok plainteks ke-2: 010110

Knapsack : 1, 5, 6, 11, 14, 20

Kriptogram: $(1 \times 5) + (1 \times 11) + (1 \times 14) = 30$

Blok plainteks ke-3: 000000

Knapsack : 1, 5, 6, 11, 14, 20

Kriptogram : 0

Blok plainteks ke-4: 011000

Knapsack : 1, 5, 6, 11, 14, 20

Kriptogram : $(1 \times 5) + (1 \times 6) = 11$

Jadi, cipherteks yang dihasilkan: 32 30 0 11

Cara membuat kunci publik dan kunci privat:

1. Tentukan barisan superincreasing.
2. Kalikan setiap elemen di dalam barisan tersebut dengan ***n*** modulo ***m***. Modulus ***m*** seharusnya angka yang lebih besar daripada jumlah semua elemen di dalam barisan, sedangkan pengali ***n*** seharusnya tidak mempunyai faktor persekutuan dengan ***m***.
3. Hasil perkalian akan menjadi kunci publik sedangkan barisan superincreasing semula menjadi kunci privat.

Enkripsi

- ✓ Enkripsi dilakukan dengan cara yang sama seperti algoritma knapsack sebelumnya.
- ✓ Mula-mula plainteks dipecah menjadi blok bit yang panjangnya sama dengan kardinalitas barisan kunci publik.
- ✓ Kalikan setiap bit di dalam blok dengan elemen yang berkoresponden di dalam kunci publik

Contoh enkripsi

- ✓Plainteks: 011000110101101110
- ✓N=6.
- ✓Kunci publik: 62, 93, 81, 88, 102, 37
- ✓Plainteks dibagi menjadi blok yang panjangnya 6, kemudian setiap bit di dalam blok dikalikan dengan elemen yang berkorepsonden di dalam kunci publik.

Blok plainteks ke-1 : 011000
Kunci publik : 62, 93, 81, 88, 102, 37
Kriptogram : $(1 \times 93) + (1 \times 81) = 174$

Blok plainteks ke-2 : 110101
Kunci publik : 62, 93, 81, 88, 102, 37
Kriptogram : $(1 \times 62) + (1 \times 93) + (1 \times 88) + (1 \times 37) = 280$

Blok plainteks ke-3 : 101110
Kunci publik : 62, 93, 81, 88, 102, 37
Kriptogram : $(1 \times 62) + (1 \times 81) + (1 \times 88) + (1 \times 102) = 333$

Jadi, cipherteks yang dihasilkan : 174, 280, 333

Sumber: Rinaldi Munir

Dekripsi

- ✓ Dekripsi dilakukan dengan menggunakan kunci privat.
- ✓ Mula-mula penerima pesan menghitung n–1 , yaitu balikan ***n*** modulo ***m***, sedemikian sehingga $n \cdot n^{-1} \equiv 1 \pmod{m}$.
- ✓ Kalikan setiap kriptogram dengan n–1 mod m, lalu nyatakan hasil kalinya sebagai penjumlahan elemen-elemen kunci privat untuk memperoleh plainteks dengan menggunakan algoritma pencarian solusi superincreasing knapsack.

Daftar Pustaka

- https://www.unisbank.ac.id/ojs/index.php/fti2/article/view/288
- Veronica Lusiana, Wiwien Hadikurniawati, Kriptografi Kunci Publik, 2010, DINAMIKA INFORMATIKA – Vol II No 1.
- https://docplayer.info/37588712-Sistem-kriptografi-kunci-publik.html
- https://docplayer.info/45911948-Implementasi-kriptografi-kunci-publik-dengan-algoritma-rsa-crt-pada-aplikasi-instant-messaging.html
- https://media.neliti.com/media/publications/144706-ID-implementasi-algoritma-kriptografi-rsa-u.pdf

- https://media.neliti.com/media/publications/174360-ID-pengamanan-dokumen-menggunakan-metode-rs.pdf

