Implementasi Algoritma RSA dalam Enkripsi dan Dekripsi File Teks

M. Malik I. Baharsyah - 13521029[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13521029@std.stei.itb.ac.id

***Abstract*—Keamanan suatu data sudah sewajarnya menjadi fokus utama masyarakat, khususnya data pribadi. Seiring dengan bernilainya data, kebocoran data semakin menjadi mimpi buruk bagi pemilik data. Kebocoran data dapat disebabkan oleh banyak hal, salah satunya adalah keamanan yang tidak terlalu ketat. Semakin ketat keamanan data, maka semakin sulit untuk diketahui pembobol, atau biasa disebut sebagai *hacker*. Sebagai upaya pengamanan data, salah satunya dapat menggunakan implemntasi Algoritma RSA. Melalui pendekatan Algoritma RSA, sebuah plainteks dapat dienkripsi menjadi chiperteks dan hanya dapat didekripsi dengan kunci privat yang dimiliki pengguna. Dengan memanfaatkan algortima tersebut, data semakin sulit untuk dibobol dan pengguna tidak perlu melalui mimpi buruk kebocoran data.**

***Keywords*—Algoritma RSA, Kriptografi, Enkripsi, Dekripsi.**

I. PENDAHULUAN

Keberhargaan suatu data sudah seharusnya menjadi fokus utama masyarakat, terutama di era yang sudah sangat maju sekarang ini. Seiring dengan bernilainya data, kebocoran data semakin menjadi mimpi buruk bagi pemilik data. Kebocoran data dapat disebabkan oleh banyak hal, salah satunya adalah keamanan yang tidak terlalu ketat. Semakin ketat keamanan data, maka semakin sulit untuk diketahui pembobol, atau biasa disebut sebagai *hacker*. Oleh karena itu, adanya keamanan data menjadi salah satu nilai penting agar tidak terjadi kebocoran data.

Salah satu cara untuk mengamankan data adalah dengan membuat algoritma yang dapat mengenkripsi sekaligus mendekripsi data tersebut, salah satunya adalah Algoritma RSA. RSA merupakan algoritma kriptografi asimetri, yang berarti kunci yang digunakan untuk enkripsi berbeda dengan kunci yang digunakan untuk dekripsi. Dikembangkan pada tahun 1976 oleh tiga orang, yakni Ron Rivest, Adi Shamir, dan Leonard Adleman, Algoritma RSA dikenal sebagai kriptografi kunci-public (*public-key cryptography)*. RSA telah digunakan untuk transmisi data yang aman dengan teknik mengenkripsi informasi yang dikirim dalam bentuk chiperteks yang tidak dapat dipahami sehingga hanya orang dengan kunci privatnya yang dapat memulihkan informasi data tersebut. Dengan memodifikasi Algoritma RSA, proses enkripsi dan dekripsi file berupa teks dapat dilakukan.

A. *Tujuan*

Merancang dan membangun purwarupa yang mampu melakukan enkripsi dan dekripsi melalui pendekatan Algoritma RSA.

II. TEORI DASAR

A. *Pembagi Bersama Terbesar (PBB)*

Pembagi Bersama Terbesar (PBB) dari a dan b adalah bilangan bulat terbesar, misal x, sehingga x habis membagi a dan x habis membagi b. Sebagai contoh, PBB(80,12) = 4, karena 4 merupakan bilangan terbesar yang dapat membagi habis nilai 80 dan 12.

B. *Relatif Prima*

Dua buah bilangan bulat, misal a dan b, disebut relatif prima jika nilai PBB(a,b) = 1.

C. *Aritmatika Modulo*

Modulo berarti sisa hasil bagi. Sebagai contoh, misal a adalah bilangan bulat dan m adalah bilangan bulat di mana a dan m lebih besar dari 0. Operasi a mod m (dibaca "a modulo m") menghasilkan sisa pembagian a dengan m. Sebagai contoh, misal a = 14 dan m = 3. Maka nilai a mod m adalah 14 dibagi 3 sisa 2.

D. *Kongruen*

Dua buah bilangan bulat, misal a dan b, dikatakan kongruen apabila dibagi bilangan bulat lain, misal m, menghasilkan sisa pembagian yang sama. Kongruen disimbolkan dengan (≡). Penulisan kongruen biasanya diikuti oleh modulo, yakni $a \equiv b \pmod{m}$ (dibaca " a kongruen b dalam modulo m") yang berarti a menghasilkan nilai yang sama dengan b jika dilakukan operasi modulo m.

E. *Kriptografi*

Kriptografi Kriptografi adalah ilmu yang mempelajari cara untuk menjaga suatu informasi tetap aman dengan cara menyandikan informasi tersebut sehingga sulit dibaca atau dimengerti oleh pihak yang tidak berhak. Pesan yang dirahasiakan dinamakan plainteks, sedangkan pesan hasil penyandian disebut cipherteks. Untuk mengembalikan pesan yang sudah disandikan ke pesan semula, seseorang harus mengetahui metode penyandian dan kunci penyandian pesan

tersebut sehingga hanya orang yang berhak yang bisa mengembalikannya ke keadaan awal. Proses menyandikan plainteks menjadi cipherteks disebut enkripsi, sedangkan proses membalikkan cipherteks menjadi plainteks disebut dekripsi. Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

1. Kerahasiaan (confidentiality) Kerahasiaan adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.

2. Integritas data Integritas adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihakpihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

3. Autentikasi Autentikasi adalah berhubungan dengan identifikasi/pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.

4. Non-repudiasi atau nirpenyangkalan Non-repudiasi atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

F. *Kriptografi RSA*

Sandi RSA merupakan algoritma kriptografi kunci publik (asimetris). Ditemukan pertama kali pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Len Adleman. Nama RSA sendiri diambil dari ketiga penemunya tersebut. Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat). Untuk menemukan kunci dekripsi, dilakukan dengan memfaktorkan suatu bilangan bulat menjadi faktor-faktor primanya. Kenyataannya, memfaktorkan bilangan bulat menjadi faktor primanya bukanlah pekerjaan yang mudah. Karena belum ditemukan algoritma yang efisien untuk melakukan pemfaktoran. Cara yang bisaa digunakan dalam pemfaktoran adalah dengan menggunakan pohon faktor. Jika semakin besar bilangan yang akan difaktorkan, maka semakin lama waktu yang dibutuhkan. Jadi semakin besar bilangan yang difaktorkan, semakin sulit pemfaktorannya, semakin kuat pula algoritma RSA.

G. *Algoritma RSA*

Algoritma ditemukan pada tahun 1976 oleh tiga peneliti yang berasal dari Massacussets Institute of Technology, yaitu Ronald Rivest, Adi Shamir, dan Leonard Adleman. Dalam Algoritma RSA, setiap pengguna memiliki sepasang kunci, yaitu kunci publik, *e*, yang berguna untuk mengenkripsi pesan dan kunci privat, *p*, yang berguna untuk mendekripsi pesan. Kunci publik bersifat publik dan kunci privat bersifat rahasia, hanya pengguna yang memiliki kunci privat. Dalam Algoritma RSA dapat dibagi menjadi dua aksi, yaitu enkripsi dan dekripsi. Proses enkripsi adalah proses mengubah teks yang dapat dipahami menjadi teks yang tidak dapat dipahami. Proses dekripsi adalah proses mengubah teks terenkripsi menjadi teks yang dapat dipahami, tetapi dalam proses dekripsi hanya dapat dilakukan oleh pemilik kunci privat.

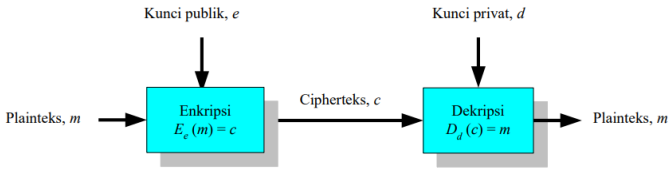
Proses enkripsi dalam Algoritma RSA yaitu:

1. Memilih dua bilangan prima, *p* dan *q*, yang bersifat rahasia.
2. Menghitung $n = p * q$, nilai n tidak perlu dirahasiakan.
3. Menghitung m = $(p – 1) (q – 1)$, bersifat rahasia.
4. Memilih sebuah bilangan bulat untuk kunci publik, *e*, yang relatif prima terhadap *m*, yaitu PBB(*e*, *m*) = 1.
5. Menghitung kunci dekripsi, *d*, dengan *d* didapat dari $ed \equiv 1 \pmod{m}$.
6. *c* = p^e mod n, dengan c adalah hasil enkripsi.

Proses dekripsi dalam Algoritma RSA yaitu:

1. p = c^d mod n, dengan p adalah hasil dekripsi.

Ilustrasi Algoritma RSA digambarkan seperti pada Gambar 1.



Gambar 1. Ilustrasi Algoritma RSA
Sumber: *Munir, Rinaldi, 2022. Bahan Kuliah IF2120 Matematika Diskrit: Teorema Bilangan (Bag. 3)*

H. *Sistem ASCII*

Plain teks yang akan dienkripsi dengan RSA Coding merupakan angka-angka, sedangkan pesan yang dikirimkan bisaanya berbentuk teks atau tulisan. Sehingga dibutuhkan suatu kode yang sifatnya universal untuk mengubah pesan teks menjadi plain teks dalam bentuk bilangan. ASCII (American Standard Code for Information Interchange) atau Kode Standar Amerika untuk pertukaran informasi merupakan suatu standar internasional dalam kode huruf dan symbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". ASCII selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks.

I. *Kekuatan dan Keamanan RSA*

Penemu algoritma RSA menyarankan nilai p dan q panjangnya lebih dari 100 digit. Dengan demikian hasil kali r = p × q akan berukuran lebih dari 200 digit. Menurut Rivest dan kawan-kawan, uasaha untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik). Inilah yang membuat algoritma RSA tetap dipakai hingga saat

ini. Selagi belum ditemukan algoritma yang mangkus untuk memfaktorkan bilangan bulat menjadi faktor primanya, maka algoritma RSA tetap direkomendasikan untuk enkripsi pesan.

III. RSA DALAM ENKRIPSI TEKS

Dalam impelentasi Algoritma untuk enkripsi teks, teks terlebih dahulu diubah ke dalam bentuk ASCII, yang merupakan angka, lalu diberlakukan Algoritma RSA. Hasil enkripsi merupakan angka, karena plainteks yang awalnya berbentuk teks telah diubah ke dalam bentuk ASCII.

IV. IMPLEMENTASI

Pada makalah ini, penulis menggunakan bahasa Python untuk mengimplementasikan algoritma enkripsi dan dekripsi teks dengan RSA. Alasan penulis menggunakan Python adalah karena Python merupakan salah satu bahasa yang mudah dimengerti. Berikut adalah penjelasan fungsi dan prosedur serta algoritma utama yang diimplementasikan dalam bahasa Python.

A. Import library

Dengan melakukan import library random dan os, dapat difungsikan untuk mencari nilai random *p* dan *q* untuk bilangan prima.

```
import random
import os
```

Gambar 2. Import library

B. Fungsi pbb

Fungsi pbb diperlukan untuk mencari nilai pembagi bersama terbesar dari bilangan bulat *a* dan *b*.

```
# fungsi untuk mencari nilai PBB(a,b)
def pbb(a, b):
    while b != 0:
        a, b = b, a % b
    return a
```

Gambar 3. Fungsi pbb

C. Fungsi isPrime

Fungsi isPrime diperlukan untuk mengecek sebuah angka apakah bilangan prima atau bukan.

```
# fungsi untuk mengecek apakah bilangan prima atau ti
def isPrime(n):
    if n == 2:
        return True
    if n % 2 == 0 or n <= 1:
        return False

    sqr = int(n**0.5) + 1

    for divisor in range(3, sqr, 2):
        if n % divisor == 0:
            return False
    return True
```

Gambar 4. Fungsi isPrime

D. Fungsi mod_inverse

Fungsi mod_inverse digunakan untuk mencari nilai inverse modulo

```
# fungsi untuk mencari nilai invers modulo
def mod_inverse(a, m):
    for x in range(1, m):
        if (a * x) % m == 1:
            return x
    return None
```

Gambar 5. Fungsi mod_inverse

E. Fungsi generate_pq

Fungsi generate_pq digunakan untuk menentukan bilangan prima acak *p* dan *q*.

```
# random p dan q bilangan prima
def generate_pq():
    while True:
        p = random.randrange(1, 1000)
        if isPrime(p):
            break
    while True:
        q = random.randrange(1, 1000)
        if isPrime(q):
            break
    return p, q
```

Gambar 6. Fungsi generate_pq

F. Fungsi generate_key

Fungsi generate_key digunakan untuk mencari kunci publik dan kunci privat.

```
# fungsi mereturn kunci publik dan kunci privat
def generate_key(p, q):
    n = p * q
    m = (p - 1) * (q - 1)
    e = random.randrange(1, m)
    g = pbb(e, m)
    while True:
        e = random.randrange(1, m)
        g = pbb(e, m)
        d = mod_inverse(e, m)
        if g == 1 and e != d:
            break
    d = mod_inverse(e, m)
    return (e, n), (d, n)
```

Gambar 7. Fungsi generate_key

G. Fungsi encrypt

Fungsi encrypt digunakan untuk mengenkripsi teks.

```
# fungsi untuk mengenkripsi teks
def encrypt(en, plaintext):
    e, n = en
    cipher = [(ord(char) ** e) % n for char in plaintext]
    return cipher
```

Gambar 8. Fungsi encrypt

H. Fungsi decrypt

Fungsi decrypt digunakan untuk mendekripsi hasil enkripsi teks.

```
# fungsi untuk mendekripsi teks
def decrypt(dn, ciphertext):
    d, n = dn
    plain = [chr((char ** d) % n) for char in ciphertext]
    return ''.join(plain)
```

Gambar 9. Fungsi decrypt

I. Fungsi programUtama

Fungsi programUtama digunakan untuk menyatukan semua fungsi menjadi satu kesatuan.

```
# fungsi utama yang menyatukan semua fungsi
def programUtama():
    print("Pilih tujuan Anda:\n 1. Enkripsi\n 2. Dekripsi")
    pil = int(input("Masukkan pilihan Anda: "))
    while pil not in [1, 2]:
        print("Pilihan tidak valid")
        pil = int(input("Masukkan pilihan Anda: "))

    if pil == 1:
        p, q = generate_pq()
        print("Hasil nilai p dan q (bilangan prima):")
        print("p = ", p)
        print("q = ", q)
        public, private = generate_key(p, q)
        print("Kunci publik Anda (tidak bersifat rahasia): ", public)
        print("Kunci privat Anda (mohon simpan dengan baik): ", private)
        print("Pilih input teks Anda:\n 1. File\n 2. Ketik sendiri")
        pil = int(input("Masukkan pilihan Anda: "))
        while pil not in [1, 2]:
            print("Pilihan tidak valid")
            pil = int(input("Masukkan pilihan Anda: "))
        if pil == 1:
            path = os.getcwd()
            files = os.listdir(path + "\\files")
            namafile = input("Masukkan nama file Anda yang terlatak di folder bernama \"files\": ")
            while namafile not in files:
                print("File tidak ditemukan")
                namafile = input("Masukkan nama file Anda yang terlatak di folder bernama \"files\": ")
            namafile = path + "\\files\\" + namafile
            f = open(namafile, "r")
            text = f.read()
            f.close()
            print("Teks yang akan dienkripsi:\n", text)
        else:
            text = input("Masukkan pesan untuk dienkripsi:\n")
        encrypted_text = encrypt(public, text)
        print("Teks hasil enkripsi:\n", ' '.join(str(c) for c in encrypted_text))
    else:
        d = int(input("Masukkan d dalam kunci privat Anda (d,n): "))
        n = int(input("Masukkan n dalam kunci privat Anda (d,n): "))
        private = (d, n)
        pil = int(input("Pilih input teks Anda:\n 1. File\n 2. Ketik sendiri\nMasukkan pilihan A
        while pil not in [1, 2]:
            print("Pilihan tidak valid")
            pil = int(input("Masukkan pilihan Anda: "))
        if pil == 1:
            path = os.getcwd()
            files = os.listdir(path + "\\files")
            namafile = input("Masukkan nama file Anda yang terlatak di folder bernama \"files\"
            while namafile not in files:
                print("File tidak ditemukan")
                namafile = input("Masukkan nama file Anda yang terlatak di folder bernama \"file
            namafile = path + "\\files\\" + namafile
            f = open(namafile, "r")
            encrypted_text = f.read()
            f.close()
            print("Teks yang akan didekripsi:\n", encrypted_text)
        else:
            encrypted_text = input("Masukkan pesan hasil enkripsi: ")
        encrypted_text = encrypted_text.split(' ')
        encrypted_text = list(map(int, encrypted_text))
        print("Teks hasil dekripsi :\n", decrypt(private, encrypted_t
```

Gambar 10. Fungsi programUtama

V. PERCOBAAN

Setelah implementasi program algoritma RSA selesai, perlu dilakukan percobaan untuk mengetes.

A. Menu Utama

Pada menu utama, pengguna akan diminta untuk memilih salah satu, yaitu proses enkripsi atau dekripsi.

```
Pilih tujuan Anda:
 1. Enkripsi
 2. Dekripsi
Masukkan pilihan Anda:
```

Gambar 11. Menu Utama

B. Proses Enkripsi

Pada saat pengguna memilih untuk melakukan enkripsi, akan muncul nilai p dan q, kunci publik, dan kunci privat.

```
Pilih tujuan Anda:
 1. Enkripsi
 2. Dekripsi
Masukkan pilihan Anda: 1
Hasil nilai p dan q (bilangan prima):
p =  997
q =  251
Kunci publik Anda (tidak bersifat rahasia):  (164671, 250247)
Kunci privat Anda (mohon simpan dengan baik):  (25231, 250247)
Pilih input teks Anda:
 1. File
 2. Ketik sendiri
Masukkan pilihan Anda:
```

Gambar 12. Proses Enkripsi

C. Hasil Enkripsi Teks

Setelah pengguna memilih input plainteks, maka akan muncul hasil enkripsi teks tersebut.

```
Masukkan nama file Anda yang terlatak di folder bernama "files": plain1.txt
Teks yang akan dienkripsi:
 ak syg km
tpi ap km syg ak?
Teks hasil enkripsi:
 57641 121920 80352 111618 205363 115442 80352 121920 105993 2692 92005 56262 119434 80352 57641 56262 80352 121920 1059
93 80352 111618 205363 115442 80352 57641 121920 176458
Apakah Anda ingin mengulang program? (y/n)
Masukkan pilihan Anda:
```

Gambar 13. Hasil Enkripsi Teks

D. Proses Dekripsi

Ketika pengguna memilih dekripsi, akan diminta kunci privat yang didapat ketika berhasil melakukan enkripsi.

```
Pilih tujuan Anda:
 1. Enkripsi
 2. Dekripsi
Masukkan pilihan Anda: 2
Masukkan d dalam kunci privat Anda (d,n): 130237
Masukkan n dalam kunci privat Anda (d,n): 238753
Pilih input teks Anda:
 1. File
 2. Ketik sendiri
Masukkan pilihan Anda:
```

Gambar 14. Proses Dekripsi

E. Hasil Proses Dekripsi

Setelah pengguna memasukkan input dekripsi, maka akan muncul hasil dekripsi dengan kunci privat yang dimiliki.

```
Pilih tujuan Anda:
 1. Enkripsi
 2. Dekripsi
Masukkan pilihan Anda: 2
Masukkan d dalam kunci privat Anda (d,n): 130237
Masukkan n dalam kunci privat Anda (d,n): 238753
Pilih input teks Anda:
 1. File
 2. Ketik sendiri
Masukkan pilihan Anda: 1
Masukkan nama file Anda yang terlatak di folder bernama "files": chiper1.txt
Teks yang akan didekripsi:
 58136 29886 209196 116121 205052 214576 209196 29886 93485 47895 120968 78430 147237 209196 58136 78430 209196 29886 93
485 209196 116121 205052 214576 209196 58136 29886 81026
Teks hasil dekripsi :
 ak syg km
tpi ap km syg ak?
Apakah Anda ingin mengulang program? (y/n)
Masukkan pilihan Anda:
```

Gambar 15. Hasil Proses Dekripsi

VI. KESIMPULAN

Algoritma RSA mampu mengamankan sebuah data bertipe teks dengan mengubahnya terlebih dahulu ke bentuk ASCII lalu dilakukan enkripsi. Ketika didapat kunci privat, pengguna harus menyimpannya dan bersifat rahasia agar dapat mendekripsi hasil enkripsi yang telah dihasilkan dan tidak diketahui publik. Algoritma RSA merupakan salah satu algoritma yang sulit untuk dilakukan pembobolan.

VII. UCAPAN TERIMA KASIH

Penulis panjatkan puji syukur kehadirat Allah SWT karena atas rahmat-Nya penulis dapat menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada keluarga yang selalu mendukung penulis selama proses penulisan. Akhir kata, penulis mengucapkan terima kasih kepada semua dosen pengampu mata kuliah Matematika Diskrit, terutama Bapak Dr.

Ir. Rinaldi Munir, M.T. selaku dosen K3 atas ilmu yang telah diberikan kepada penulis sehingga penulis dapat mennyelesaikan makalah ini.

REFERENCES

[1] informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-2021/Teori-Bilangan-2020-Bagian3.pdf diakses pada 12 Desember 2021 pukul 19.00.WIB.
[2] https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2022-2023/matdis22-23.htm#Makalah diakses pada 12 Desember 2021 pukul 19.00.WIB.
[3] https://www.tutorialkart.com/python/python-read-file-as-string/ diakses pada 12 Desember 2021 pukul 19.00.WIB.
[4] https://payahtidur.com/project/rsa diakses pada 12 Desember 2021 pukul 19.00.WIB.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Desember 2022

M. Malik I. Baharsyah 13521029