

**Tugas 4 IF4020 Kriptografi
Semester II Tahun 2023/2024**

Serangan terhadap RSA / RSA Attack

Deadline : Kamis, 18 April 2024
Tempat pengumpulan : <https://forms.gle/uwVu3dsaTbYREkUu9>
Berkas pengumpulan : File format PDF
Anggota kelompok : 2-3 orang
QnA :
<https://docs.google.com/spreadsheets/d/1csV5V3yBy5a8KoUETKMduP8B0gwJEff7vt31KtqzbBk/edit?usp=sharing>

Bagian A



Setelah kejadian pada tugas 2 lalu, Conan menyadari bahwa surat percakapan antara dirinya dengan Ai Haibara dapat dipecahkan oleh Ayumi Yoshida. Oleh sebab itu, Conan yang sudah mempelajari kriptografi kunci publik beralih ke algoritma RSA dalam merahasiakan pesan yang hendak dikirim ke Ai Haibara dan juga sebaliknya. Namun selayaknya manusia yang tidak luput akan kesalahan, ternyata Conan lalai dalam mengkonfigurasi properti-properti dari RSA yang ia gunakan untuk mengenkripsi pesan. Hal ini tentu akan berakibat fatal sebab pesan cipher akan lebih mudah untuk dipecahkan.

Ayumi Yoshida menyadari bahwa Conan sudah tidak lagi menggunakan Hill Cipher. Termakan api cemburu, ia mengerahkan segala upaya untuk mengetahui isi surat yang dipertukarkan antara Conan dan Ai Haibara. Sejauh ini, ia telah berhasil mendapatkan akses ke *channel* yang digunakan Conan dan Ai Haibara dalam bertukar pesan cipher. Lebih lanjut, ia juga telah sukses dalam mendapatkan kode Python yang digunakan untuk mengenkripsi dengan RSA. Berikut adalah kode yang dimaksud:

```
from Crypto.Util.number import *
import random
from sympy import nextprime

flag = "RAHASIA"
tahap = 30
paket_soal = ["A", "B", "C", "D", "E"]

print(f"Selesaikan {tahap} tahap untuk mendapatkan flag!\n")
print("Kirimkan plainteks dalam bentuk format KRIPTOGRAFIITB{secret}!\n")
print("Tips: buatlah kode untuk otomasi :D\n")
counter = 0
try:
    for step in range(tahap):
        print(f"----- Tahap-{step}-----\n")
        message_asli = "KRIPTOGRAFIITB{" + str(random.randint(1,10000))+ "}"
        message_asli = message_asli.encode('utf-8')
        message_int = bytes_to_long(message_asli)
        version = random.choice(paket_soal)
        print(f"paket_soal = {version}\n")
        if version == "A":
            while True:
                ran = random.randint(1, 100)
                p = nextprime(getStrongPrime(1024) - ran)
                q = nextprime(nextprime(nextprime(nextprime(p) + ran) + ran) - ran)
                n = p * q
                e = 65537
                check = GCD(e, (p-1)*(q-1)) == 1
                if check: break
            enc = pow(message_int, e, n)
        elif version == "B":
            p = getStrongPrime(1024)
            n = p * p
            e = 65537
            enc = pow(message_int, e, n)
        elif version == "C":
            while True:
                p = getStrongPrime(1024)
                q = getStrongPrime(1024)
                e = random.randrange(1,65537)
                n = p * q
                tot = (p-1) * (q-1)
                e = random.randint(2**15, 2**16)
                check = GCD(e, (p-1)*(q-1)) == 1
                if check: break
            d = pow(e, -1, tot)
```

```

        enc = pow(message_int, d, n)
        e = d
    elif version == "D":
        p = getStrongPrime(1024)
        q = getStrongPrime(1024)
        n = p*q
        e = 3
        enc = pow(message_int, e, n)
    elif version == "E":
        n = getStrongPrime(1024)
        e = 65537
        enc = pow(message_int, e, n)

    print(f"n = {n}\n")
    print(f"e = {e}\n")
    print(f"c = {enc}\n")

    try:
        print("Jawaban = ")
        input_dec = input().strip("\n")
        if input_dec == message_asli.decode():
            print("Uwaw keren!!!\n")
            counter += 1
        else:
            print(":((((((\n")
    except Exception as e:
        print("Error\n")
except Exception as e:
    print("Error\n")
finally:
    if counter == tahap:
        print(f"Uhuyyyy {flag}\n")
    else:
        print("Tetap semangat dan jangan putus asa!\n")

```

Tak mau kalah dari Conan dan Ai Haibara, Ayumi Yoshida juga turut mempelajari algoritma RSA. Ia menyadari bahwa RSA yang dipakai Conan lemah apabila ditinjau dari nilai-nilai p, q, e, dan n yang dipakai. Ayumi Yoshida lalu meminta bantuan Anda sebagai ahli IT untuk memecahkan pesan-pesan cipher dari RSA lemah tersebut.

Tugas Anda adalah

1. Amati kode yang dipakai Conan di atas. Perhatikan bahwa ada lima kasus (A-E) konfigurasi yang membuat RSA Conan lemah.
2. Untuk setiap kasus, jelaskan secara detail di laporan hal-hal yang membuat RSA untuk kasus tersebut lemah.
3. Untuk setiap kasus, jelaskan secara detail di laporan langkah-langkah serangan yang dapat dilakukan untuk mendapatkan plainteks dari cipherteks jika diketahui nilai n, e, dan c.
4. Buatlah sebuah kode Python untuk melakukan serangan sesuai dengan apa yang Anda jelaskan untuk poin 3.

5. Uji kode yang telah Anda buat dengan cara berikut:

a. Gunakan netcat untuk mengakses nilai n, e, dan c yang tersedia pada

host: **165.232.161.196**

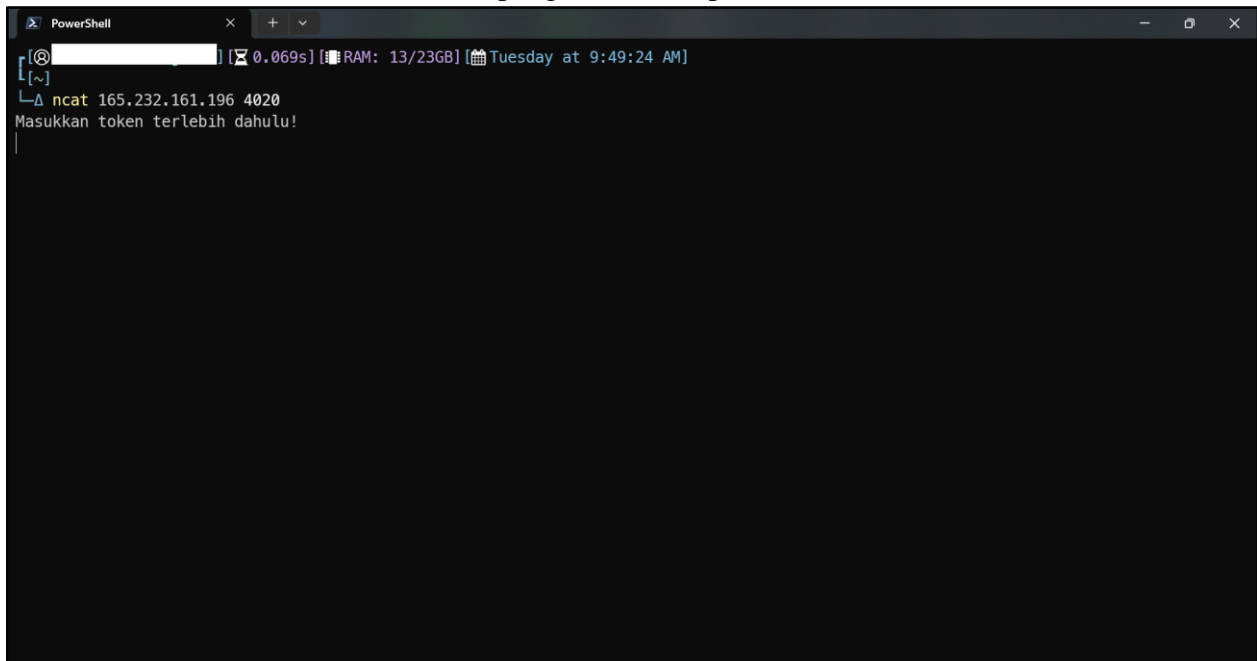
port: **4020**

Contoh:

```
## Pada Windows  
ncat 165.232.161.196 4020
```

```
## Pada Linux dan Mac  
nc 165.232.161.196 4020
```

b. Setelah terhubung ke server, Anda akan diminta untuk menginput sebuah token. Silakan masukkan token yang diberikan untuk kelompok Anda. Catatan: Token akan diberikan melalui email setelah deadline pengisian kelompok.



```
PowerShell  
[C:\>] [0.069s] [RAM: 13/23GB] [Tuesday at 9:49:24 AM]  
[~]  
ncat 165.232.161.196 4020  
Masukkan token terlebih dahulu!  
|
```

c. Setelah memasukkan token, Anda akan disajikan 30 (tiga puluh / *trente*) kasus uji yang masing-masing memiliki nilai n, e, c, dan versi kasus yang digunakan.

```
PowerShell
[~] 0.069s [RAM: 13/23GB] [Tuesday at 9:49:24 AM]
ncat 165.232.161.196 4020
Masukkan token terlebih dahulu!
Selesaikan 30 tahap untuk mendapatkan flag!
Kiriman plainteks dalam bentuk format KRIPTOGRAFIITB{secret}!
Tips: buatlah kode untuk otomasi :D
----- Tahap-0-----
paket_soal = C
n = 239756006115827235968401136772116726591038207943922239682716247245121412420158112745457266789826186922158198381244537738226127693
4907479046323720728486691701264213918610113406809161836821960995608075973568748912176130844191374957638160912368212790925462725519982
7529949735723286176714464618099810992529554289650229771746221372647442088356275840806572109085248742981937251994968677393544093534999
9647012950158959216089491491936110644994665663290799768191270431364299400816441269159489991214953860520413521225602455230861913379716
61209486600980354979431045161470951405664701826015541163155786800761828884574823050088423
e = 122958806120440747423805730890329977799923856848982980184074173078671394145919371315208909667636905486788317855447964665133349181
9000269698303617340464875408402196706364255476706612607380363308941379016224144084216966125429479971174883707843004751858995633497508
4045329114880746933651107301892631213004447123887928894546759237753668957173362366863313196851650686231225301908730577077373818757486
8744495563114279415300469548104867960436443361991831727299266146617027354446289566132187864776397810941339303193659084007992233921239
83617966666389395634554396627883695223422655605293708778329882606372404497175684459392721
c = 72782400243795695575972321974269499002373729917053804075091881586982056688858517938878022103137509222079282922442615160856466820
1961485350441068054990440220295890633622899043749259449275147175460474115677926655969455755491661648527125205656587513988908934676954
2205073113749704588899510458998748759853588815961751350885801199962703769016727504883218974849656647705291803674014337078602139139022
6504808412061560251935872150921741743974844876006992463918519619170971387948996957317492274669207500903795796638356506265547642297869
5956215266932641880415365004994981179755116382142598239172571604048521799212745287444009
Jawaban = |
```

- d. Silahkan pecahkan nilai m menggunakan kode yang telah anda buat. Sebelum dimasukkan sebagai jawaban. Ubah m yang Anda dapatkan dari tipe long ke bentuk bytes untuk mendapatkan jawaban dalam format **b'KRIPTOGRAFIITB{secret}'**. Lalu, Masukkan bagian **KRIPTOGRAFIITB{secret}** sebagai jawaban.
- e. Setelah berhasil menjawab seluruh kasus uji dengan benar, Anda akan menerima sebuah flag dalam format **Uhuyyyy KRIPTOGRAFIITB{secret}**. Silakan cantumkan flag tersebut ke laporan.
- f. Ambil satu contoh test case untuk masing-masing kasus (A-E). Jelaskan pada laporan terkait bagaimana program anda menyelesaikan test case tersebut.
- g. **Kode disimpan dalam GitHub dan pranala dicantumkan dalam laporan**

Bagian B



Kriptografi ITB membuat sebuah arsip digital yang dapat digunakan untuk menyimpan suatu informasi. Pengguna dapat memasukkan nomor arsip yang diinginkan, kemudian nomor arsip tersebut akan dienkripsi menggunakan RSA untuk menghasilkan suatu akses token yang rahasia. Untuk melakukan akses ke suatu arsip, pengguna perlu memasukkan token rahasia, kemudian program akan mendekrip token tersebut dan melakukan akses ke nomor arsip berdasarkan token yang didekrip.

Sonoko melamar kerja di Kriptografi ITB sebagai admin. Dia menyimpan suatu rahasia pada arsip tersebut yang tidak ingin diketahui oleh orang lain. Conan memberi tahu Sonoko bahwa arsip digital pada Kriptografi ITB tersebut tidaklah aman jika orang lain mengetahui nomor arsip dimana Sonoko menyimpan rahasianya. Sonoko kemudian memberikan tantangan kepada Conan untuk membaca isi dari pesan rahasia yang dia simpan pada nomor arsip admin. Bantulah Conan menemukan isi pesan rahasia tersebut! Sebagai hint, Conan diperbolehkan melihat kode arsip digital.

```
from Crypto.Util.number import *
import random

flag = "RAHASIA"

def main():
    arsip = {}
    p = getStrongPrime(512)
    q = getStrongPrime(512)
    n = p * q
    tot = (p-1) * (q-1)
    e = random.randint(2**15, 2**16)
    while GCD(e, tot) != 1:
        e = random.randint(2**15, 2**16)
    d = pow(e, -1, tot)

    nomor_arsip_admin = random.randint(2, 100000000)
    while isPrime(nomor_arsip_admin):
        nomor_arsip_admin = random.randint(2, 100000000)

    arsip[nomor_arsip_admin] = flag

    try:
        perintah = "0"
        while perintah != "4":
            print("Selamat datang di arsip digital Kriptografi ITB!")
            print("Ketik angka untuk menjalankan perintah: ")
            print("1. Tambah Arsip")
            print("2. Baca Arsip")
            print("3. Nomor Arsip Admin")
            print("4. Keluar")
            print("")
            print("Masukkan perintah: ", end="")
            perintah = input().strip("\n")
```

```

if perintah == "1":
    print("Masukkan nomor arsip (dalam bentuk integer): ")
    nomor_arsip = input().strip("\n")
    if int(nomor_arsip) == int(nomor_arsip_admin):
        print("Nomor arsip admin tidak boleh diganti")
        continue
    print("Masukkan isi arsip: ", end="")
    input_arsip = input().strip("\n")
    arsip[int(nomor_arsip)] = input_arsip
    cipher_nomor_arsip = pow(int(nomor_arsip), e, n)
    print(f"Token akses nomor arsip: {cipher_nomor_arsip}")
elif perintah == "2":
    try:
        print("Masukkan token akses nomor arsip (dalam bentuk integer): ", end="")
        nomor_arsip = input().strip("\n")
        plain_nomor_arsip = pow(int(nomor_arsip), d, n)
        print(f"Isi arsip: {arsip[int(plain_nomor_arsip)]}")
    except Exception as e:
        print("Token akses nomor arsip invalid")
elif perintah == "3":
    print(f"Nomor arsip admin: {nomor_arsip_admin}")
elif perintah == "4":
    exit()
else:
    print("Perintah tidak dikenal")
except Exception as e:
    print("Terjadi kesalahan")

if __name__ == "__main__":
    main()

```

Tugas Anda adalah

1. Akses arsip digital Kriptografi ITB menggunakan netcat ke
host: **165.232.161.196**
port: **1303**
2. Sama seperti sebelumnya, Anda akan diminta untuk memasukkan token terlebih dahulu.
3. Setelah memasukkan token Anda akan disajikan sebuah menu dengan empat pilihan.

```
PowerShell root@ubuntu-s-1vcpu-2gb-70 [0.025s] [RAM: 14/23GB] [Tuesday at 10:51:43 AM]
[~]
ncat 165.232.161.196 1303
Masukkan token terlebih dahulu!

Selamat datang di arsip digital Kriptografi ITB!
Ketik angka untuk menjalankan perintah:
1. Tambah Arsip
2. Baca Arsip
3. Nomor Arsip Admin
4. Keluar

Masukkan perintah:
```

4. Jika memilih “3. Nomor Arsip Admin”, anda akan diberikan sebuah angka yang menunjukkan nomor arsip milik admin.

```
PowerShell root@ubuntu-s-1vcpu-2gb-70 [0.025s] [RAM: 14/23GB] [Tuesday at 10:51:43 AM]
[~]
ncat 165.232.161.196 1303
Masukkan token terlebih dahulu!

Selamat datang di arsip digital Kriptografi ITB!
Ketik angka untuk menjalankan perintah:
1. Tambah Arsip
2. Baca Arsip
3. Nomor Arsip Admin
4. Keluar

Masukkan perintah: 3
Nomor arsip admin: 28685962
Selamat datang di arsip digital Kriptografi ITB!
Ketik angka untuk menjalankan perintah:
1. Tambah Arsip
2. Baca Arsip
3. Nomor Arsip Admin
4. Keluar

Masukkan perintah:
```

5. Jika memilih “1. Tambah Arsip”, Anda akan diminta memasukkan nomor untuk arsip Anda beserta isi dari arsip tersebut. Kemudian arsip akan tersimpan dan Anda akan menerima sebuah token akses arsip. Perlu diingat, token akses arsip berasal dari nomor arsip yang dienkripsi menggunakan algoritma RSA. Selain itu, Anda tidak bisa memasukan nomor arsip admin sebagai nomor arsip baru.


```
PowerShell root@ubuntu-s-1vcpu-2gb-70
3. Nomor Arsip Admin
4. Keluar

Masukkan perintah: 3
Nomor arsip admin: 28685962
Selamat datang di arsip digital Kriptografi ITB!
Ketik angka untuk menjalankan perintah:
1. Tambah Arsip
2. Baca Arsip
3. Nomor Arsip Admin
4. Keluar

Masukkan perintah: 1
Masukkan nomor arsip (dalam bentuk integer): 6789
Masukkan isi arsip: Ligma
Token akses nomor arsip: 10109263384027037354525607183967548239325800952991932536706258317917602907143553663033288725
730702316125013464640070566940337506810413942293554068957486037873793364486362604437704587235411997125213965769560092
8333450196753692220736483800502621028227984692177631594431599124111965916990563228518227081071480702
Selamat datang di arsip digital Kriptografi ITB!
Ketik angka untuk menjalankan perintah:
1. Tambah Arsip
2. Baca Arsip
3. Nomor Arsip Admin
4. Keluar

Masukkan perintah:
```

6. Jika memilih “2. Baca Arsip”, Anda akan diminta memasukkan token akses arsip untuk membaca isi arsip.

```
PowerShell root@ubuntu-s-1vcpu-2gb-70
Masukkan nomor arsip (dalam bentuk integer): 6789
Masukkan isi arsip: Ligma
Token akses nomor arsip: 10109263384027037354525607183967548239325800952991932536706258317917602907143553663033288725
730702316125013464640070566940337506810413942293554068957486037873793364486362604437704587235411997125213965769560092
8333450196753692220736483800502621028227984692177631594431599124111965916990563228518227081071480702
Selamat datang di arsip digital Kriptografi ITB!
Ketik angka untuk menjalankan perintah:
1. Tambah Arsip
2. Baca Arsip
3. Nomor Arsip Admin
4. Keluar

Masukkan perintah: 2
Masukkan token akses nomor arsip (dalam bentuk integer): 101092633840270373545256071839675482393258009529919325367062
583179176029071435536630332887257307023161250134646400705669403375068104139422935540689574860378737933644863626044377
045872354119971252139657695600928333450196753692220736483800502621028227984692177631594431599124111965916990563228518
227081071480702
Isi arsip: Ligma
Selamat datang di arsip digital Kriptografi ITB!
Ketik angka untuk menjalankan perintah:
1. Tambah Arsip
2. Baca Arsip
3. Nomor Arsip Admin
4. Keluar

Masukkan perintah: |
```

7. Tujuan Anda adalah membaca isi dari arsip milik admin. Untuk itu tulislah secara jelas di laporan Anda:
- Permasalahan yang ada
 - Dasar Teori Penyelesaian Masalah

- c. Langkah-langkah yang Anda lakukan untuk menyelesaikan masalah. Sertakan *screenshot*
- d. Isi dari arsip admin.

Catatan

1. Buatlah sebuah kelompok beranggotakan 2-3 orang. Tulis nama anggota kelompok pada sheets:
<https://docs.google.com/spreadsheets/d/1busxpgDMwxme4ZsOTJNqbuBAc4pCe1qb7CckufXIcjA/edit?usp=sharing>
Maksimal pengisian adalah **Kamis, 4 April 2024 sebelum jam kuliah.**
2. Format laporan bebas, pastikan penjelasan Anda logis dan mudah dimengerti.
3. Mohon untuk tidak melakukan hal aneh-aneh kepada server demi kebaikan bersama :D.
4. Anda diperbolehkan memanfaatkan library untuk kode serangan. Tetapi tidak boleh library yang fungsinya adalah mengcrack RSA.
5. Untuk bagian A, disarankan melakukan pengujian secara lokal terlebih dahulu dengan menyesuaikan dan menggunakan kode yang telah diberikan.
6. Kumpulkan laporan dengan format nama Tugas4_<Nomor Kelompok> disesuaikan dengan nomor di sheets daftar kelompok. Contoh: Tugas4_Z.pdf

Referensi

1. <https://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf>
2. https://www.cs.purdue.edu/homes/jblocki/courses/555_Spring17/slides/Lecture31.pdf
3. <https://www.nku.edu/~christensen/Mathematical%20attack%20on%20RSA.pdf>