Analisis Kerentanan Implementasi RSA dengan Eksponen Privat Rendah Menggunakan Wiener's Attack dan Pecahan Berlanjut

Muhammad Abduh - 13525077
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: mabduh1357@gmail.com , 13525077@std.stei.itb.ac.id

***Abstract*—The RSA algorithm is a widely utilized asymmetric cryptographic system that secures modern digital communication by relying on the computational difficulty of factoring large integers. However, specific implementation practices can introduce fatal vulnerabilities. A common scenario involves selecting a small private exponent d to accelerate the decryption process, particularly in resource-constrained environments. This paper investigates the vulnerability of RSA cryptosystems when implemented with a low private exponent, specifically focusing on Wiener's Attack. By utilizing the mathematical principles of continued fractions and Diophantine approximations, Wiener demonstrated that if the private exponent satisfies the condition $d < \frac{1}{3}N^{\frac{1}{4}}$, the system can be compromised in polynomial time. The attack exploits the convergence properties of continued fractions to extract the private key directly from the public key parameters without factoring the modulus. Through theoretical analysis and programmatic simulation, this paper validates the mathematical boundaries of vulnerability and details the extraction mechanisms. The findings reinforce the necessity of proper parameter selection and recommend alternative optimization techniques, such as the Chinese Remainder Theorem, to achieve decryption without compromising cryptographic integrity.**

Keywords—RSA; Wiener's attack' continued fractions; low private exponent cryptanalysis.

I. PENDAHULUAN

Di era digital seperti saat ini, komunikasi secara daring (komunikasi digital) telah menjadi hal yang tidak terpisahkan dari kehidupan sehari-hari manusia. Sama halnya dengan komunikasi secara langsung, komunikasi digital memerlukan kepastian bahwa hanya pihak-pihak yang ikut berkomunikasi yang dapat mengetahui isi dari komunikasi tersebut. Hal ini menjadi sebuah tantangan karena proses komunikasi secara daring memiliki banyak celah yang dapat dimanfaatkan oleh pihak yang tidak berwenang. Mengingat pentingnya menjaga privasi pengguna pada komunikasi digital, diperlukan sebuah sistem yang dapat memberikan keamanan dan kepastian mengenai kerahasiaan pesan bagi pengguna.

Salah satu sistem yang sering digunakan adalah kriptografi berbasis algoritma RSA. RSA telah menjadi standar de facto untuk enkripsi asimetris dan tanda tangan digital sejak pertama kali diperkenalkan. Keamanan algoritma ini bersandar pada asumsi matematis bahwa memfaktorkan bilangan bulat yang sangat besar menjadi faktor primanya adalah permasalahan yang sangat sulit dikomputasi.

Meskipun tanggung secara teoritis, impementasi RSA di dunia nyata seringkali berhadapan dengan kendala komputasi. Proses dekripsi maupun pembuatan tanda tangan digital yang melibatkan operasi perpangkatan modulo dengan eksponen besar membutuhkan waktu yang signifikan. Dalam praktik, untuk mempercepat komputasi pada perangkat keras dengan sumber daya terbatas(seperti *smart card* atau perangkat IoT), sering terjadi pengurangan eksponen privat d yang berukuran kecil [1]. Praktik ini sangat berbahaya karena reduksi ruang pencarian dari d membuka celah eksploitasi matematis yang fatal terhadap kerahasiaan kunci privat secara keseluruhan [7].

Sebagai sebuah solusi analitik terhadap celah ini, M.J. Wiener pada tahun 1990 memperkenalkan Wiener;s Attack [1]. Serangan ini mengeksploitasi hubungan matematis antara kuci publik dan kunci privat dengan menggunakan algoritma pecahan berlanjut (*continued fractions)* untuk memecahkan RSA dalam waktu polinomial. Wiener membuktikan bahwa kunci privat *d* dapat dipulihkan secara penuh dengan komputasi efisien jika ukuran $d < \frac{1}{3}N^{\frac{1}{4}}$. Penemuan ini memicu eksplorasi lanjutan di bidang kriptoanalisis, termasuk perbaikan batas keamanan kerentanan RSA menjadi $N^{0.292}$oleh Boneh dan Durfee pada tahun 200 menggunakan metode reduksi basis lattice [3].

Makalah ini bertujuan untuk menganalisis kerentanan RSA dengan eksponen privat rendah secara mendalam. Selain itu, makalah ini juga bertujuan untuk membuktikan batas kerentatan Wiener secara matematis.

II. TEORI DASAR

A. Konsep Dasar RSA

RSA atau Rivest-Shamir-Adleman adalah metode enskripsi asimetris yang menggunakan sepasang kunci berbeda yaitu kunci publik yang berfungsi untuk mengenkripsi *sebuah plain text* menjadi *cipher text* dan kunci privat yang berfungsi untuk mendekripsi sebauh *cipher text* menjadi *plain text* [2].

Algoritma RSA didasarkan pada teorema Euler yang menyatakan bahwa

$$a^{\phi(n)} \equiv 1 (\text{mod n}) \quad (1)$$

dengan keterangan,

1. a harus relatif prima terhadap n
2. $\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \ldots \left(1 - \frac{1}{p_r}\right)$, dimana p_1, p_2, ..., p_r adalah faktor prima dari n.

$\phi(n)$ atau *Euler's Totient Function* adalah fungsi yang menentukan berapa banyak dari bilangan-bilangan 1,2,3, …, n yang relatif prima terhadap n.

Berdasarkan sifat $a^k \equiv b^k \pmod{n}$ untuk k bilangan bulat ≥ 1 maka persamaan (1) dapat ditulis menjadi

$$a^{k\phi(n)} \equiv 1^k \pmod{n} \; atau \; a^{k\phi(n)} \equiv 1 \pmod{n} \quad (2)$$

Bila a diganti dengan m, maka persamaan (2) menjadi

$$m^{\phi(n)} \equiv 1 \pmod{n} \quad (3)$$

Berdasarkan sifat $ac \equiv bc \pmod{n}$, maka persamaan (3) dikali m menjadi

$$m^{k\phi(n)+1} \equiv m \pmod{n} \quad (4)$$

dimana m relatif prima terhadap n.

Misalkan e dan d dipilih sedemikian sehingga

$$e \cdot d \equiv 1 \left(\text{mod } \phi(n)\right) \; atau \; e \cdot d = k\phi(n) + 1 \quad (5)$$

Sulihkan (5) ke dalam persamaan (4) menjadi:

$$m^{e \cdot d} \equiv m \pmod{n} \; atau \; (m^e)^d \equiv m \pmod{n} \quad (6)$$

yang artinya, perpangkatan m dengan e lalu dipangkatkan lagi dengan d menghasilkan m semula.

Berdasarkan persamaan (6), maka enkripsi dan dekripsi dirumuskan sebagai berikut :

$$E_e(m) = c \equiv m^e \pmod{n} \quad (7)$$

$$D_d(c) = m \equiv c^d \pmod{n} \quad (8)$$

Proses pembangkitan kunci RSA terdiri dari beberapa komputasi sekuensial. Pertama, dipilih dua bilangan prima rahasia yang bernilai besar, yaitu p dan q. Modulus RSA kemudian dihitung sebagai hasil kali dari kedua bilangan prima tersebut, yaitu $n = p \times q$. Selanjutnya, dihitung fungsi totient Euler dengan rumus $\phi(n) = (p-1)(q-1)$. Kunci publik e dipilih sedemikian rupa sehingga berada pada rentang $1 < e < \phi(n)$ dan memenuhi syarat gcd(e, $\phi(n)$) = 1. Langkah terakhir adalah menghitung kunci privat d yang merupakan invers perkalian modular dari e sehingga memenuhi relasi $e \cdot d \equiv 1 \left(\text{mod } \phi(n)\right)$.

Setelah pasangan kunci terbentuk, proses enkripsi terhadap pesan m menjadi ciphertext c dilakukan menggunakan persamaan $c = m^e \pmod{n}$. Untuk mengembalikan pesan asli, penerima melakukan proses dekripsi dengan mengeksekusi $m \equiv c^d \pmod{n}$.

Sebagai contoh numerik sederhana, misalkan dipilih p = 61 dan q = 53. Modulus yang dihasilkan adalah n = 3233, dengan $\phi(n)$ = 3120. Jika dipilih e = 17, maka d yang memenuhi $17\ d \equiv 1 \pmod{3120}$ adalah d = 2753. Pesan m = 65 akan dienkripsi menjadi $c = 65^{17} \pmod{3223} = 2790$ dan dapat didekripsi kembali melalui $65 \equiv 2790^{2753} \pmod{3233}$.

B. Algoritma Euclidian & Aritmatika Modulo

Aritmatika modulo adalah sistem operasi matematika bilangan bulat di mana nilai-nilainya berputar kembali setelah mencapai suatu batas yang disebut modulus. Relasi kekongruenan dinotasikan sebagai $a \equiv b \pmod{n}$ yang bermakna bahwa sisa pembagian a oleh n sama dengan sisa pembagian b oleh n.

Salah satu konsep fundamental dalam aritmatika modulo adalah *Greatest Commond Divisior* (GCD) atau Pembagi Bersama Terbesar (PBB). Algoritma Euclidian menyediakan metode rekursif yang sangat efisien untuk menghitung GCD dari dua bilangan bulat tanpa perlu melakukan faktorisasi. Algoritma ini berjalan dengan secara berulah membagi bilangan yang lebih besar dengan bilangan yang lebih kecil lalu menggantikan bilangan pembagi dengan sisa bagi hingga sisa bagi tersebut bernilai nol. Pembagi tak nol terakhir adalah nilai GCD. Sebagai contoh, untuk mencari gcd(120,35), langkah yang dilakukan adalah

$$120 = 3 \times 35 + 15$$
$$35 = 2 \times 15 + 5$$
$$15 = 3 \times 5 + 0$$

karena sisa terakhir tak nol adalah 5, maka gcd(120,35) = 5.

C. Pecahan Berlanjut (Continued Fractions)

Pecahan berlanjut adalah representasi matematika dari sebuah bilangan riil α dalam bentuk rangkaian penjumlahan dan pembagian bersarang. Secara formal, ekspansi pecahan berlanjut ditulis sebagai:

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

Ekspresi tersebut umumnya dipersingkat menjadi notasi barisan kuosien barisan $\alpha = [a_0; a_1, a_2, a_3, \ldots]$. Nilai a_0 merepresentasikan bagian bulat $\lfloor \alpha \rfloor$, sementara $a_1, a_2, a_3, \ldots$ diekstrak secara rekursif dari pembalikan sisa pecahannya [5]. Aprosimaksi rasional terbaik dari α yang dipotong pada suku ke-n disebut sebagai konvergen dan dinotasikan sebagai fraksi p_n/q_n. Nilai pembilang dan penyebut konvergen ini dapat dihitung menggunakan relasi rekurensi

$$p_n = a_n p_{n-1} + p_{n-2}$$

$$q_n = a_n q_{n-1} + q_{n-2}$$

dengan kondisi awal $p_1 = 1, p_0 = a_0\ serta\ q_1 = 0, q_0 = 1$.

Kekuatan analisis pecahan berlanjut terletak pada Teorema Legendre yang berbunyi: Jika terdapat sebuah pecahan rasional p/q sedemikian rupa sehingga jarak aproksimasinya terhadap α memenuhi kondisi $\left|\alpha - \frac{p}{q}\right| < \frac{1}{2q^2}$ maka dapat dipastikan bahwa p/q adalah salah satu konvergen dari ekspansi pecahan berlanjut α. Teorema inilah yang menjadi fondasi utama dalam eksploitasi eksponen privat. Sebagai contoh, ekspansi pecahan berlanjut dari aproksimasi $17993/90581$ akan menghasilkan kuosien parsial $[0; 5, 30, 2, 1, 3, 2, \dots]$, yang secara bertahap membentuk deret konvergen.

III. ANALISIS WIENER'S ATTACK

A. Teorema Wiener

Teorema Wiener yang dipublikasikan pada tahun 1990 menyatakan bahwa: Misalkan $N = pq$ di mana ukuran q dan p berada dalam batas $q < p < 2q$. Apabila sebuah eksponen privat d dipilih secara rentan sedemikian sehingga $d < \frac{1}{3}N^{\frac{1}{4}}$, maka jika diberikan pasangan kunci publik (N, e) dengan relasi $e \cdot d \equiv 1 \pmod{\phi(n)}$, nilai d dapat diekstraksi secara efisien dalam waktu polinomial menggunakan ekspansi pecahan berlanjut dari parameter publik e/N [1].

Motivasi di balik penyerangan ini berakar pada struktur persamaan Diophantine $e \cdot d - k\phi(n) = 1$. Jika d bernilai sangat kecil, pembagian persamaan tersebut akan membuktikan bahwa fraksi tak diketahui k/d memiliki selisih nilai yang sangat tipis dengan parameter publik e/N. Analisis matematis membuktikan bahwa ketika batasan $d < \frac{1}{3}N^{\frac{1}{4}}$terpenuhi, jarak kesalahannya akan tertekan hingga $\left|\frac{e}{N} - \frac{k}{d}\right| < \frac{1}{2d^2}$. Berkat jaminan dari Teorema Legendre, fraksi k/d secara mutlak akan muncul di antara daftar konvergen hasil ekspansi pecahan berlanjut nilai e/N [6], [8].

B. Reduksi Persamaan Diophantine

Derivasi matematis Wiener's Attack dibangun melalui serangkaian reduksi aljabar yang elegan.

Langkah 1 berawal dari definisi kunci RSA, yakni relasi kongruensi $e \cdot d \equiv 1 \pmod{\phi(n)}$. Relasi ini dapat diekspresikan sebagai persamaan kesetaraan linier dengan sebuah bilangan bulat k, menjadi $e \cdot d = k\phi(n) + 1$, yang setara dengan

$$e \cdot d - k\phi(n) = 1$$

Langkah 2 adalah melakukan aproksimasi nilai $\phi(n)$ menggunakan N. Berdasarkan definisi, $\phi(n) = (p-1)(q-1) = N - p - q + 1$. Mengingat syarat operasional $q < p < 2q$, penjumlahan $p + q$dibatasi pada batas atas $3\sqrt{N}$. Karenanya, besaran galat dapat dihitung sebagai:

$$|N - \phi(N)| = p + q - 1 < 3\sqrt{N}$$

Pertidaksamaan ini menunjukkan bahwa $\phi(N)$adalah pendekatan yang sangat dekat untuk N apabila N bernilai besar.

Langkah 3 melibatkan pembagian persamaan pada Langkah 1 dengan bilangan dN, sehingga memunculkan variabel aproksimasi e/Ndan k/d:

$$\left|\frac{e}{N} - \frac{k}{d}\right| = \frac{|ed - kN|}{dN} = \frac{|k\phi(N) + 1 - kN|}{dN}$$

$$\left|\frac{e}{N} - \frac{k}{d}\right| = \frac{|k(\phi(N) - N) + 1|}{dN}$$

Berdasarkan fakta bahwa $|N - \phi(N)| < 3\sqrt{N}$dan bahwa nilai $k < d$(karena $e < \phi(N)$) , maka pembilangnya terikat oleh batasan:

$$\left|\frac{e}{N} - \frac{k}{d}\right| < \frac{3k\sqrt{N}}{dN} < \frac{3d\sqrt{N}}{dN} = \frac{3}{\sqrt{N}}$$

Langkah 4 berfokus pada aplikasi Teorema Legendre. Berdasarkan premis Wiener, jika syarat eksponen $d < \frac{1}{3}N^{1/4}$ dipenuhi, maka kuadrat dari $d$adalah $d^2 < \frac{1}{9}N^{1/2}$, yang dapat ditulis sebagai $3\sqrt{N} < \frac{N}{3d^2}$. Melalui reduksi batas atas,

kondisi ini menyebabkan:

$$\left|\frac{e}{N} - \frac{k}{d}\right| < \frac{1}{2d^2}$$

Terpenuhinya pertidaksamaan ini menyimpulkan bahwa rahasia rasional k/dtidak perlu ditebak, melainkan cukup dihitung deterministik sebagai salah satu konvergen dari e/N [1].

C. Ekstraksi Konvergen

Proses ekstraksi dilakukan dengan sebuah algoritma sistematis yang memanen konvergen dan mengujinya.

1. Hitung seluruh daftar konvergen dari ekspansi pecahan berlanjut kunci publik$e/N = [a_0; a_1, a_2, \dots]$.
2. Iterasi untuk setiap konvergen k/ddari daftar tersebut. Lakukan validasi berikut:
    a. Verifikasi relasi modulo: periksa apakah $k$dapat membagi habis $(ed - 1)$.
    b. Jika pembagian menghasilkan bilangan bulat tak bersisa, hitung kandidat $\phi(N) = \frac{ed-1}{k}$.
    c. Gunakan kandidat $\phi(N)$untuk menyusun persamaan kuadrat: $x^2 - (N - \phi(N) + 1)x + N = 0$. Akar dari persamaan kuadrat ini adalah$p$dan q.
    d. Apabila nilai diskriminan dari persamaan kuadrat tersebut menghasilkan akar berupa bilangan bulat sempurna, maka nilai d yang sedang diuji adalah kunci privat yang absolut benar.

3. Jika seluruh iterasi konvergen gagal menemukan akar bulat, maka dijamin eksponen d tidak memenuhi kriteria kerentanan $\frac{1}{3}N^{1/4}$.

Kompleksitas algoritma pencarian ini sangat efisien dan berjalan secara polinomial dalam lingkup $O(\log^2 N)$[4]. Efisiensi ini terjadi karena jumlah elemen parsial dalam pecahan berlanjut sangat terbatas, dan setiap tahapnya hanya memerlukan pengujian akar polinomial derajat dua sederhana [9].

Sebagai studi kasus numerik untuk membedah iterasi ekstraksi ini secara komprehensif, mari kita asumsikan sebuah sistem RSA memiliki parameter modulus $N = 90581$dan eksponen publik $e = 17993$ [1]. Langkah pertama yang dieksekusi oleh penyerang adalah menjabarkan parameter e/Nke dalam struktur algoritma Euclidean untuk menemukan kuosien parsialnya. Penjabaran matematis baris demi barisnya adalah sebagai berikut:

$$90581 = 5 \times 17993 + 616$$

$$17993 = 29 \times 616 + 129$$

$$616 = 4 \times 129 + 100$$

$$129 = 1 \times 100 + 29$$

$$100 = 3 \times 29 + 13$$

TABEL I Iterasi Ekspansi Konvergen Dari Fraksi $\frac{17993}{90581}$

Iterasi (n)	Kuosien (an)	Pembilang (pn)	Penyebut (qn)	Konvergen (k/d)
0	0	0	1	0 / 1
1	5	1	5	1 / 5
2	29	29	145	29 / 145
3	4	117	585	117 / 585

Proses pengujian dilakukan mulai dari iterasi pertama ($n = 1$). Pada iterasi ini, sistem mengambil kandidat $k = 1$ dan $d = 5$. Relasi divisibilitas diuji menggunakan perhitungan $\phi(N) = (17993 \times 5 - 1)/1 = 89964$. Kandidat fungsi Euler ini kemudian disubstitusikan ke dalam persamaan akar kuadrat untuk memecah modulus N:

$$x^2 - (90581 - 89964 + 1)x + 90581 = 0$$

$$x^2 - 618x + 90581 = 0$$

Menghitung diskriminan dari persamaan tersebut menghasilkan akar-akar rasional bulat yang bernilai $p = 379$ dan $q = 239$. Keberadaan akar riil yang bulat ini memvalidasi secara absolut bahwa tebakan kunci privat $d = 5$adalah valid. Serangan dihentikan hanya pada iterasi kedua dengan tingkat keberhasilan mutlak, mendemonstrasikan betapa mematikannya efisiensi reduksi pecahan berlanjut ini..

IV. IMPLEMENTASI DAN SIMULASI

A. *Skenario Simulasi*

Untuk membuktikan batas matematis secara empiris, implementasi komputasi dilakukan menggunakan perangkat lunak berbasis bahasa pemrograman Python 3. Evaluasi simulasi dirancang ke dalam tiga buah skenario komputasi berdasarkan variasi rentang nilai eksponen privat. Skenario 1 merupakan simulasi sistem rentan, di mana kunci diatur sedemikian rupa agar panjang `bernilai sangat kecil ($d < N^{1/4}$). Skenario 2 merupakan titik batas (boundary), menguji arsitektur dengan nilai $d \approx N^{1/4}$. Skenario 3 adalah arsitektur RSA yang ideal (Aman), di mana panjang kunci $d$diposisikan cukup besar, yakni melampaui separuh dari modulus N.Pengujian dieksekusi dengan menetapkan ukuran modulus N pada 512-bit. Modulus berukuran kecil ini dipilih untuk memastikan metrik evaluasi dapat menangkap respons kecepatan (execution time) dalam skala milidetik dan berfokus pada pembuktian ekstraksi d.

B. *Implementasi Kode*

Arsitektur algoritma solver kriptanalisis ini dipecah menjadi tiga fungsi struktural.

Fungsi pertama berfungsi mengubah parameter publik menjadi pecahan berlanjut. Algoritma ini berjalan selayaknya rekursi Euclidean.

```python
def continued_fraction(numerator, denominator):
    quotients = []
    while denominator:
        quotients.append(numerator // denominator)
        numerator, denominator = denominator, numerator % denominator
    return quotients
```

Gambar 4.1 Implementasi fungsi untuk mengubah parameter publik menjadi pecahan berlanjut
(Sumber: arsip pengguna)

Fungsi kedua memanfaatkan persamaan rekurensi untuk menyusun daftar rasio konvergen p_n/q_n.

```python
def convergents(quotients):
    convergent_list = []
    p_prev, p_curr = 1, quotients[0]
    q_prev, q_curr = 0, 1
    convergent_list.append((p_curr, q_curr))
    for a in quotients[1:]:
        p_prev, p_curr = p_curr, a * p_curr + p_prev
        q_prev, q_curr = q_curr, a * q_curr + q_prev
        convergent_list.append((p_curr, q_curr))
    return convergent_list
```

Gambar 4.2 Implementasi fungsi untuk menyusun daftar rasio konvergen p_n/q_n
(Sumber: arsip pengguna)

Fungsi inti ketiga mensimulasikan mekanisme Wiener's Attack untuk mencari akar dari diskriminan persamaan kuadrat $x^2 - (N - \phi(N) + 1)x + N = 0$.

```python
def wiener_attack(e, N):
    quotients = continued_fraction(e, N)
    convs = convergents(quotients)
    for k, d in convs:
        if k == 0 or (e * d - 1) % k != 0:
            continue
        phi_N = (e * d - 1) // k
        b = N - phi_N + 1
        discriminant = b * b - 4 * N
        if discriminant >= 0:
            sqrt_disc = int(discriminant ** 0.5)
            if sqrt_disc * sqrt_disc == discriminant:
                return d # Serangan berhasil
    return None
```

Gambar 4.3 Implementasi fungsi untuk mencari akar dari diskriminan persamaan kuadrat $x^2 - (N - \phi(N) + 1)x + N = 0$
(Sumber: arsip pengguna)

Arsitektur dari ketiga fungsi di atas merangkum sebuah diagram alur logika yang sepenuhnya otonom. Saat parameter publik e dan N disuntikkan ke dalam sistem, program tidak melakukan metode brute-force yang memakan kompleksitas waktu $O(2^N)$, melainkan mendelegasikan tugas tersebut pada pencarian topologi konvergen. Algoritma continued_fraction bertindak sebagai generator state, mereduksi ruang pencarian miliaran kombinasi menjadi kurang dari seratus daftar kuosien. Selanjutnya, fungsi wiener_attack bertindak sebagai validator state.

Optimalisasi pada baris kode validasi diskriminan (sqrt_disc * sqrt_disc == discriminant) sengaja dirancang untuk menghindari penggunaan pustaka floating-point yang kompleks. Dengan memaksa tipe data tetap berada di dalam rentang integer, program secara drastis memangkas alokasi siklus CPU, mengizinkan penyerang untuk meretas parameter enkripsi dengan latensi waktu dalam skala pecahan milidetik.

C. *Hasil Eksekusi*

Tabel II memvisualisasikan matriks respons algoritma pengujian terhadap kerentanan dari masing-masing arsitektur modulus.

Tabel II Hasil Eksekusi Wiener's Attack Pada Rsa 512-Bit

Skenario	Ukuran d	d<N1/4	Hasil Serangan	Waktu (ms)
1 (Rentan)	32-bit	Ya	Berhasil	< 1
2 (Batas)	128-bit	Tidak	Kadang Berhasil	2
3 (Aman)	200-bit	Tidak	GAGAL	-

Untuk memvalidasi tabel hasil di atas serta mendemonstrasikan mekanisme audit keamanan secara *real-time*, pengujian dilakukan secara langsung menggunakan antarmuka baris perintah (*Command Line Interface*). Skrip *solver* Python dieksekusi secara lokal untuk merekam jejak pencarian konvergen dan waktu pemrosesan CPU (*elapsed time*).

Berikut adalah log keluaran (*output log*) dari terminal saat skrip serangan Wiener dijalankan pada ketiga arsitektur RSA 512-bit tersebut:

```
> python3 tes.py
============================================================
       [ LOG EKSEKUSI WIENER'S ATTACK - RSA 512-BIT ]
============================================================

[*] Menjalankan SKENARIO 1 (Sistem Rentan - d 32-bit)...
    Parameter N : 57353148261275530553664794886625471...
    Parameter e : 18545623521544124468927204525781264...
    Batas Teorema : d harus < 9173140826899205819674048311660118016O (approx 1/3 N^0.25)
    [!] Mengekstrak pecahan berlanjut dan konvergen...
    [+] STATUS: BERHASIL DITEMBUS!
    [+] Kunci Privat (d) terekstraksi: 2431537249
    [*] Waktu Eksekusi: 0.3853 ms
------------------------------------------------------------
[*] Menjalankan SKENARIO 2 (Ambang Batas - d 128-bit)...
    Parameter N : 4801111032813537676722976401349406 9...
    Parameter e : 3523811059757714287895143066638856 7...
    Batas Teorema : d harus < 8774334417777834777022250257633981235 2 (approx 1/3 N^0.25)
    [!] Mengekstrak pecahan berlanjut dan konvergen...
    [+] STATUS: BERHASIL DITEMBUS!
    [+] Kunci Privat (d) terekstraksi: 188309498491948110053904799939525041 13
    [*] Waktu Eksekusi: 0.3235 ms
------------------------------------------------------------
[*] Menjalankan SKENARIO 3 (Sistem Aman - e standar 65537)...
    Parameter N : 880332984334923544625936836052922 60...
    Parameter e : 65537...
    Batas Teorema : d harus < 10210346693566256961455568170278243532 8 (approx 1/3 N^0.25)
    [!] Mengekstrak pecahan berlanjut dan konvergen...
    [-] STATUS: EKSPLOITASI GAGAL (Sistem Aman / Batas Meleset)
    [*] Waktu Eksekusi: 0.0224 ms
------------------------------------------------------------
                    [ PROSES SELESAI ]
============================================================
```

Gambar 4.4 Log Hasil Testing serangan Wiener Pertama
(Sumber: arsip pengguna)

```
> python3 tes.py
============================================================
       [ LOG EKSEKUSI WIENER'S ATTACK - RSA 512-BIT ]
============================================================

[*] Menjalankan SKENARIO 1 (Sistem Rentan - d 32-bit)...
    Parameter N : 11753030880021462156482801153939200...
    Parameter e : 11367692258652678394786652275552750...
    Batas Teorema : d harus < 1097529531108623698083352105569128284 16 (approx 1/3 N^0.25)
    [!] Mengekstrak pecahan berlanjut dan konvergen...
    [+] STATUS: BERHASIL DITEMBUS!
    [+] Kunci Privat (d) terekstraksi: 3265862219
    [*] Waktu Eksekusi: 0.1829 ms
------------------------------------------------------------
[*] Menjalankan SKENARIO 2 (Ambang Batas - d 128-bit)...
    Parameter N : 8955050277053768991728100064337120 5...
    Parameter e : 8252910456965676673335612107266745 0...
    Batas Teorema : d harus < 1025405759370108761199782031463945338 88 (approx 1/3 N^0.25)
    [!] Mengekstrak pecahan berlanjut dan konvergen...
    [-] STATUS: EKSPLOITASI GAGAL (Sistem Aman / Batas Meleset)
    [*] Waktu Eksekusi: 0.4838 ms
------------------------------------------------------------
[*] Menjalankan SKENARIO 3 (Sistem Aman - e standar 65537)...
    Parameter N : 1115772112613386883041274301367030 3...
    Parameter e : 65537...
    Batas Teorema : d harus < 1083359562115474802211117965205695365 12 (approx 1/3 N^0.25)
    [!] Mengekstrak pecahan berlanjut dan konvergen...
    [-] STATUS: EKSPLOITASI GAGAL (Sistem Aman / Batas Meleset)
    [*] Waktu Eksekusi: 0.0224 ms
------------------------------------------------------------
                    [ PROSES SELESAI ]
============================================================
```

Gambar 4.5 Log Hasil Testing serangan Wiener Kedua
(Sumber: arsip pengguna)

Berdasarkan log eksekusi terminal di atas, dapat diobservasi bahwa algoritma Euclidean mampu memilah fraksi yang tepat di dalam memori dan meruntuhkan sandi dalam waktu kurang dari 1 milidetik pada Skenario 1. Sebaliknya, penolakan

konvergen secara instan pada Skenario 3 memperlihatkan bahwa fungsi dekripsi dengan panjang eksponen privat yang memenuhi standar (melebihi batas ambang $N^{0.25}$) tidak akan pernah memberikan celah bagi eksploitasi berbasis aproksimasi Diophantine, dikarenakan pecahan publik e/N telah sepenuhnya kehilangan relasi proksimasi matematisnya terhadap parameter rahasia.

V. KESIMPULAN

Analisis mendalam pada studi ini membuktikan kerentanan fatal pada implementasi RSA yang memaksakan eksponen privat $d$bernilai kecil. Secara matematis, sistem kriptografi ini terbukti runtuh saat eksponen privat memenuhi batasan $d < \frac{1}{3}N^{1/4}$. Penerapan Wiener's Attack memperlihatkan bahwa melalui ekspansi pecahan berlanjut pada parameter kunci publik e/N, daftar rasio konvergen secara pasti akan mengandung kombinasi rahasia k/d.

Implikasi teoritis dari serangan ini menyoroti bahwa masalah keamanan RSA tidak semata bersandar pada faktorisasi prima, namun juga terkait erat dengan aproksimasi Diophantine. Teorema Legendre menjamin temuan akar persamaan tanpa mengharuskan penyerang memecahkan N. Celah ini menunjukkan sifat arsitektur aljabar yang dapat membocorkan state internal jika relasi $ed - k\phi(N) = 1$dieksekusi dengan variabel berukuran rendah.

Guna menangkal vektor serangan ini secara praktis, eksponen privat harus dibangkitkan sedemikian rupa agar memiliki ukuran minimal setengah panjang bit dari modulus (misalnya minimal 512-bit untuk modul RSA 1024-bit). Jika akselerasi dekripsi menjadi sebuah keharusan di tingkat perangkat keras (hardware), pengembang disarankan menerapkan optimasi menggunakan Chinese Remainder Theorem (CRT) alih-alih mengecilkan parameter d.

Pada pengembangan mendatang, batas $N^{1/4}$ini bukanlah batas kerentanan maksimal. Riset selanjutnya harus mewaspadai ekstensi serangan seperti algoritma Boneh dan Durfee [3] yang telah mendorong celah eksploitasi hingga parameter $N^{0.292}$ melalui penerapan matriks basis reduksi kisi (lattice basis reduction).

REFERENCES

[1] [1] M. J. Wiener, "Cryptanalysis of short RSA secret exponents," IEEE Trans. Inf. Theory, vol. 36, no. 3, pp. 553–558, May 1990.

[2] [2] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[3] [3] D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than N^0.292," IEEE Trans. Inf. Theory, vol. 46, no. 4, pp. 1339–1349, Jul. 2000.

[4] [4] D. Coppersmith, "Small solutions to polynomial equations, and low exponent RSA vulnerabilities," J. Cryptology, vol. 10, no. 4, pp. 233–260, 1997.

[5] [5] A. Dujella, "Continued fractions and RSA with small secret exponent," Tatra Mt. Math. Publ., vol. 29, pp. 101–112, 2004.

[6] [6] S. Maitra and S. Sarkar, "Revisiting Wiener's attack – new weak keys in RSA," in Proc. ISC 2008, Lecture Notes Comput. Sci., vol. 5222, Springer, pp. 228–243.

[7] [7] D. Boneh, "Twenty years of attacks on the RSA cryptosystem," Notices AMS, vol. 46, no. 2, pp. 203–213, 1999.

[8] [8] H.-M. Sun, W.-C. Yang, and C.-S. Laih, "On the improvement of Wiener attack on RSA with small private exponent," Sci. World J., vol. 2014, Art. no. 650537, 2014.

[9] [9] R. Steinfeld, S. Contini, H. Wang, and J. Pieprzyk, "Converse results to the Wiener attack on RSA," in Proc. PKC 2005, Lecture Notes Comput. Sci., vol. 3386, Springer, pp. 184–198.

[10] [10] J. Blömer and A. May, "A generalized Wiener attack on RSA," in Proc. PKC 2004, Lecture Notes Comput. Sci., vol. 2947, Springer, pp. 1–13.

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, 19 Juni 2025

Muhammad Abduh
13525077