Evaluasi Keamanan Protokol Single Sign-On pada Aplikasi Edunex Menggunakan Kriptografi Asimetris RSA

Reinhard Alfonzo Hutabarat – 13524056
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
Email: reinhardalfonso@gmail.com, 13524056@std.stei.itb.ac.id

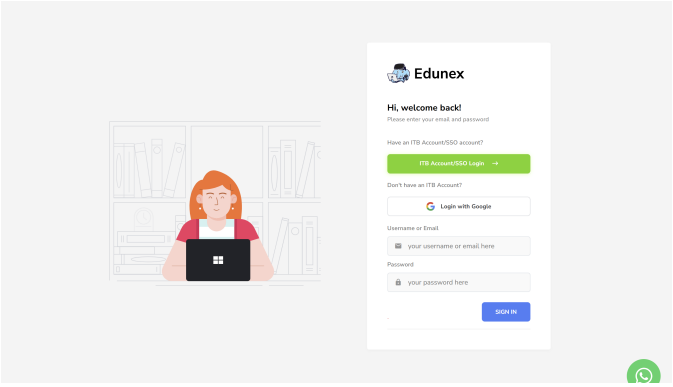
***Abstrak*—Dokumen ini menyajikan analisis konseptual mengenai potensi kerentanan keamanan pada protokol Single Sign-On (SSO) yang digunakan oleh aplikasi Edunex Institut Teknologi Bandung (ITB), dengan fokus pada peran kriptografi asimetris RSA. Makalah ini menguraikan dasar-dasar teori bilangan yang menopang algoritma RSA, termasuk konsep bilangan prima, aritmetika modular, fungsi totient Euler, dan algoritma Euclidean diperluas. Kemudian, dibahas secara rinci mekanisme kerja RSA dalam proses pembangkitan kunci, enkripsi, dan dekripsi. Selanjutnya, makalah ini mengeksplorasi Serangan Wiener, sebuah serangan terhadap RSA yang mengeksploitasi eksponen privat yang kecil, dengan menjelaskan prinsip matematisnya yang melibatkan pecahan berlanjut dan batasan keberhasilannya yang telah diperbarui. Meskipun analisis Serangan Wiener dilakukan secara konseptual dan tidak mendalam, makalah ini mengidentifikasi potensi titik kerentanan jika implementasi RSA dalam infrastruktur SSO Edunex (yang berbasis Microsoft Account/Microsoft Authenticator) menggunakan parameter kunci yang tidak optimal. Makalah ini menyimpulkan dengan menegaskan pentingnya praktik pembangkitan kunci yang kuat dan peran otentikasi multifaktor.**

***Kata Kunci*—Single Sign-On, SSO, RSA, Kriptografi Asimetris, Teori Bilangan, Serangan Wiener, Edunex, Microsoft Authenticator.**

I. PENDAHULUAN

Edunex merupakan *Learning Management System (LMS)* yang disediakan Institut Teknologi Bandung sebagai platform digital mahasiswa dalam mengakses aktivitas akademik, mulai dari distribusi materi kuliah hingga pengumpulan tugas. Dengan berbagai fitur dan manfaatnya, Edunex telah menjadi pilar dalam dunia pendidikan yang membantu menciptakan lingkungan belajar yang efektif, efisien, dan terjangkau. Salah satu metode akses utama mahasiswa ke Edunex adalah melalui ITB Account/SSO Login, yang sudah terintegrasi dengan sistem otentikasi berbasis Microsoft dan menggunakan *Multi-Factor Authentication* yaitu Microsoft Authenticator. Ketergantungan yang semakin besar pada platform digital ini menuntut jaminan keamanan yang kuat untuk melindungi data sensitif mahasiswa, menjaga integritas akademik, dan memastikan ketersediaan sistem.

Mekanisme otentikasi seperti Single Sign-On (SSO) menjadi peran yang krusial dalam postur keamanan ini, SSO memberikan kemudahan bagi para pengguna dengan memungkinkan akses ke berbagai situs atau platform menggunakan satu set kredensial, sekaligus memberikan keuntungan keamanan melalui sentralisasi manajemen identitas. Namun, pendekatan ini juga mengandung risiko penting, yaitu ketergantungan pada satu titik penyedia identitas (IdP) yang jika diserang dapat menciptakan ”titik kegagalan tunggal” dan mengganggu seluruh akses.



Gambar I.1: Tampilan login pada Edunex

(Sumber: Arsip Penulis)

Dalam konteks kemananan digital, kriptografi asimteris, khususnya algoritma RSA (Rivest-Shamir-Adleman), merupakan kunci utama yang digunakan secara luas dan masif untuk transmisi data yang aman dan tanda tangan digital. Fundamental dari algoritma RSA ini bergantung pada prinsip-prinsip teori bilangan. Meski begitu, sama halnya seperti algoritma kriptografi lainnya, implementasi RSA memiliki kerentanan terhadap serangan tertentu jika parameter kuncinya tidak dipilih dengan hati-hati. Salah satu serangan tersebut adalah *Wiener Attack's*, yang menargetkan implementasi RSA di mana eksponen privat (d) memiliki nilai yang relatif kecil.

II. Landasan Teori

A. Teori Bilangan dalam Kriptografi Asimetris RSA

1) Bilangan Prima, Pembagi Bersama Terbesar (PBB), dan Aritmetika Modulo:

- Bilangan Prima adalah bilangan bulat positif yang lebih besar dari 1 dan pembaginya hanya 1 dan dirinya sendiri. Barisan bilangan prima dimulai dari 2, yaitu 2, 3, 5, 7, 11, dan seterusnya. Dalam kriptografi, bilangan prima berfungsi sebagai blok bangunan fundamental. Keamanan algoritma RSA sangat bergantung pada kesulitan komputasi dalam memfaktorkan bilangan komposit (bilangan selain prima) yang sangat besar menjadi faktor-faktor prima penyusunnya.
- Pembagi Bersama Terbesar (PBB) dari dua atau lebih bilangan bulat adalah bilangan bulat positif terbesar yang membagi setiap bilangan tersebut tanpa sisa. Misalkan m dan n bilangan bulat, dengan syarat $n > 0$ sedemekian sehingga,

  $$m = nq + r,\ 0 \leq r < n$$

  maka PBB(m,n) = PBB(n,r).

  Contoh: Misalkan $m = 36,\ n = 24$, maka:

  $$36 = 1 \cdot 24 + 12$$

  $$\text{PBB}(36, 24) = \text{PBB}(24, 12) = 12$$

  Konsep PBB berperan aktif dalam proses pembangkitan kunci RSA untuk memastikan bahwa eksponen yang dipilih saling prima.
- Aritmetika Modulo adalah sistem aritmetika yang bekerja dengan sisa hasil pembagian untuk angka di mana angka ”berputar” setelah mencapai nilai tertentu, disebut modulus. Misalkan a dan m bilangan bulat ($m > 0$). Operasi

  $$a \bmod m$$

  memberikan sisa jika a dibagi dengan m. m disebut modulus atau modulo, dan hasil aritmetika modulo m terletak di dalam himpunan $\{0, 1, 2, \ldots, m-1\}$. Hubungan Kongruensi, yang ditulis sebagai $a \equiv b \pmod{n}$, berarti bahwa m habis membagi $(a - b)$. Aritmetika Modulo menjadi inti dari banyak operasi kriptografi, termasuk enkripsi dan tanda tangan digital. Operasi dasar seperti penjumlahan, pengurangan, dan perkalian modular memastikan bahwa hasil tetap berada dalam rentang.
  Contoh: $56 \equiv 4 \pmod{13}$.

2) Fungsi Totient Euler ($\phi(n)$):

- Totient Euler ($\phi(n)$), adalah sebuah fungsi yang menghitung jumlah bilangan bulat positif kurang dari atau sama dengan n yang relatif prima terhadap n. Dua bilangan bulat a dan b dapat dikatakan relatif prima jika PBB(a, b) = 1. Misalnya, $\phi(9) = 6$ (bilangan 1, 2, 4, 5, 7, 8). Dalam RSA, fungsi ini berguna untuk menentukan hubungan antara eksponen publik (e) dan eksponen privat (d).

3) Extended Euclidean Algorithm dan Invers Modular:

- Algoritma Euclidean tidak hanya digunakan untuk mencari PBB dari dua bilangan bulat a dan b, tetapi juga dapat mengidentifikasi kombinasi linier . Misalkan a dan b bilangan bulat positif, maka terdapat bilangan bulat m dan n sedemikian sehingga PBB(a, b) = $ma + nb$. Penerapan utamanya dalam kriptografi adalah menemukan invers modular. Bilangan d adalah invers modular dari $e \bmod n$ jika $de \equiv 1 \pmod{n}$, dan ini hanya ada jika PBB(e, n) = 1. Dalam RSA, algoritma ini sangat penting untuk menghitung eksponen privat d dari e dan $\phi(n)$, memastikan $d \equiv e^{-1} \pmod{\phi(n)}$.

4) Teorema Fermat dan Teorema Euler:

- Teorema Fermat menyatakan, jika p adalah bilangan prima dan a adalah bilangan bulat yang tidak habis dibagi dengan p, yaitu PBB(a, p) = 1, maka:

  $$a^{p-1} \equiv 1 \pmod{p}$$

- Teorema Euler adalah dasar matematis yang menjamin keberhasilan proses dekripsi pada RSA. Hal ini karena $ed \equiv 1 \pmod{\phi(n)}$ menyiratkan $ed = 1 + k\phi(n)$ untuk suatu bilangan bulat k. Dengan demikian,

  $$M^{ed} = M^{1+k\phi(n)} = M \cdot (M^k)^{\phi(n)}$$

  Berdasarkan Teorema Euler,

  $$(M^k)^{\phi(n)} \equiv 1 \pmod{n}$$

  sehingga,

  $$M^{ed} \equiv M \pmod{n}$$

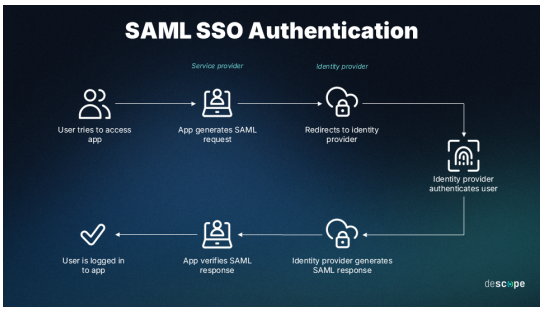
  memungkinkan pengembalian pesan asli.

B. Single Sign-On

Single Sign-On adalah metode otentikasi yang memungkinkan pengguna untuk masuk ke beberapa situs atau platform dengan satu set kredensial. Hal ini sangat memudahkan dan meningkatkan pengalaman pengguna karena pengguna tidak perlu mengingat banyak kata sandi dan melakukan operasi *login* berulang kali. Dengan Sentralisasi otentikasi, administrator dapat menerapkan langkah-langkah keamanan yang lebih kuat seperti Multi-Factor Authentication (MFA) dan praktik kata sandi yang baik secara konsisten di seluruh aplikasi. Ini mengurangi risiko terkait kata sandi yang lemah, berulang, atau hilang. Manajemen identitas terpusat memungkinkan kontrol yang lebih baik atas hak akses pengguna dan mempermudah tugas-tugas administrasi. Namun, SSO juga memiliki tantangan yang dapat merugikan pengguna. Jika server SSO atau penyedia identitas (IdP) mengalami kegagalan atau berhasil diretas, akses ke semua aplikasi yang terhubung dapat terpengaruh secara bersamaan sehingga hal ini sangat mengancam data pribadi pengguna.

Terdapat beberapa protokol yang digunakan untuk mengimplementasikan SSO, setiap protokol memiliki karakteristik dan keterbatasannya sendiri.

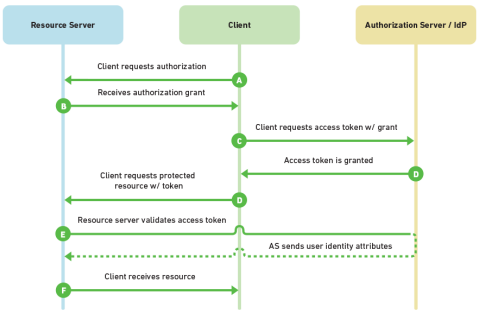
- **SAML (Security Assertion Markup Language)** adalah protokol otentikasi berbasis XML untuk manajemen identitas terfederasi. Ini memungkinkan IdP untuk memverifikasi data pengguna dan memberikan akses ke berbagai penyedia layanan (Service Provider) tanpa memerlukan *login* berulang. Berikut gambaran dari proses otentikasi SAML:



Gambar II.1: Ilustrasi proses otentikasi SSO SAML

(Sumber: https://www.descope.com/blog/post/saml-vs-sso)

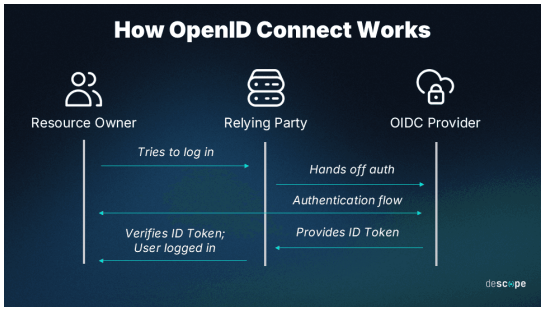
- **OAuth 2,0 (Open Authorization)** adalah protokol otorisasi, bukan otentikasi. Protokol ini sebagai penyedia akses berbasis token yang aman ke aplikasi dan API, OAuth memungkinkan untuk memberikan akses aplikasi ke sumber daya pengguna tanpa membagikan kredensial pengguna secara langsung. Protokol ini sangat berguna dalam sistem aplikasi seluler.



Gambar II.2: Ilustrasi proses otorisasi OAuth 2.0

(Sumber: https://stackoverflow.com/questions/63083666/oauth2-based-sso)

- **OpenID Connect (OIDC)** adalah ekstensi dari OAuth 2.0 yang menambahkan kemampuan. OIDC memungkinkan aplikasi untuk memverifikasi identitas pengguna dan mendapatkan informasi profil dasar pengguna dari penyedia identitas menggunakan JSON Web Tokens (JWTs). OIDC dirancang dengan beberapa fitur keamanan, yang menjadikannya protokol aman untuk autentikasi jika diterapkan dengan benar. Namun, seperti teknologi lainnya, kekuatan keamanannya sangat bergantung pada cara penerapan dan konfigurasinya.



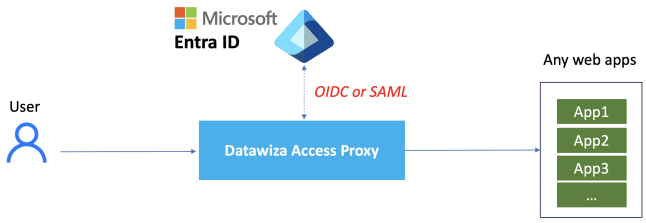
Gambar II.3: Ilustrasi proses otentikasi OpenID Connect

(Sumber: https://www.descope.com/learn/post/oidc)

C. Arsitektur SSO ITB Account

ITB Account menggunakan layanan identitas berbasis Microsoft, yaitu Microsoft Entra ID sebagai Penyedia Identitas (IdP). Microsoft Entra ID mendukung SSO berbasis federasi menggunakan protokol SAML 2.0 atau OpenID Connect.

Microsoft Authenticator berperan sebagai Multi-Factor Authentication (MFA). Hal ini berarti setelah pengguna memasukkan kredensial ITB Account mereka, mereka akan diminta untuk melakukan verifikasi tambahan melalui aplikasi Microsoft Authenticator di perangkat seluler pribadi. MFA menambahkan lapisan keamanan yang signifikan, karena jika kata sandi pengguna berhasil diretas, penyerang masih memerlukan faktor kedua seperti persetujuan di Microsoft Authenticator untuk mendapatkan akses.



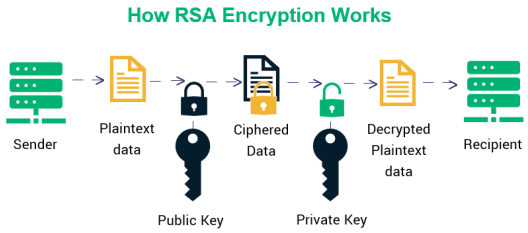
Gambar II.4: Ilustrasi Single Sign-On pada Microsoft Entra ID

(Sumber: https://www.datawiza.com/blog/)

Meskipun SSO meningkatkan keamanan secara keseluruhan, terdapat beberapa kerentanan yang dapat dieksploitasi jika tidak diimplementasikan dengan benar. Dalam konteks SSO berbasis Microsoft Entra ID, RSA digunakan untuk tujuan tanda tangan digital pada token identitas (misalnya, JWTs dalam OIDC atau SAML assertions) yang diberikan oleh IdP. Tanda tangan digital ini memastikan integritas dan keaslian token, sehingga Edunex dapat memverifikasi bahwa token tersebut benar-benar berasal dari Microsoft Entra ID dan belum dimodifikasi selama transmisi. RSA dapat melakukan pertukaran kunci yang aman atau operasi kriptografi internal lainnya dalam sistem IdP.

D. Algoritma RSA

Rivest-Shamir-Adleman merupakan tiga peneliti dari MIT (Massachussets Institute of Technology) yang membuat sebuah algoritma kriptografi asimetris yang sangat banyak digunakan. Asimetri disini artinya kunci untuk enkripsi berbeda dengan kunci untuk dekripsi, sehingga keamanannya didasarkan pada kesulitan komputasi dalam memfaktorkan bilangan bulat besar. RSA menggunakan sepasang kunci, yaitu kunci publik (e), untuk mengenkripsi data dan kunci privat (d), untuk mendekripsi data.



Gambar II.5: Alur kerja algoritma RSA

(Sumber: https://www.securew2.com/blog/what-is-rsa-asymmetric-encryption)

Proses Pembangkitan Pasangan Kunci di Dalam RSA:

1) Pemilihan dua bilangan Prima Besar p dan q:

- Pemilihan ini dilakukan secara acak dan rahasia. Bilangan prima harus sangat besar dan memiliki perbedaan yang substansial untuk memastikan kesulitan dalam faktorisasinya.

2) Perhitungan Modulus (n):

- Modulus n dihitung dengan mengalikan bilangan kedua bilangan prima yang dipilih,

  $$n = p \cdot q$$

  Nilai n akan menjadi bagian dari kunci publik dan kunci privat.

3) Perhitungan Fungsi Totient Euler:

- Nilai $\phi(n)$ dihitung menggunakan rumus,

  $$\phi(n) = (p-1)(q-1)$$

  Fungsi Totient Euler ini sangat penting karena akan menentukan grup perkalian dan menjadi modulus untuk pemilihan eksponen e dan d.

4) Pemilihan Eksponen Publik (e):

- Sebuah bilangan e dipilih sedemikan rupa sehingga $1 < e < \phi(n)$ dan PBB$(e, \phi(n))$ = 1. e dikenal sebagai eksponen enkripsi atau publik.

5) Perhitungan Eksponen Privat (d):

- Eksponen Privat d dihitung sedemikian rupa sehingga memenuhi kongruensi,

  $$ed \equiv 1 \pmod{\phi(n)}$$

  Ini berarti d adalah invers modular dari $e \bmod \phi(n)$. Perhitungan d ini dilakukan menggunakan Algoritma Euclidean Diperluas dan nilai d harus dijaga kerahasiaannya sangat ketat, karena ini adalah kunci dekripsi.

Enkripsi dilakukan dengan cara mengubah *plaintext*(p) menjadi *ciphertext* (c),

$$c = p^e \bmod n$$

Ciphertext (c) kemudian dikirimkan kepada penerima. Setelah menerima c, penerima menggunakan kunci privat mereka (n, d) untuk mendekripsi pesan dan mengembalikannya ke bentuk *plaintext* (p) asli menggunakan formula,

$$p = c^d \bmod n$$

Proses ini dinamakan juga proses dekripsi yang dijamin oleh Teorema Euler.

E. Wiener's Attack

Serangan Wiener adalah serangan terhadap RSA yang mengeksploitasi pemilihan kunci privat d yang terlalu kecil. Serangan ini berhasil ketika eksponen privat d memenuhi kondisi,

$$d < \frac{1}{3}\sqrt[4]{N},$$

di mana N adalah modulus RSA.

Kita tahu bahwa kongruensi $ed \equiv 1 \pmod{\phi(n)}$ dapat ditulis sebagai persamaan:

$$ed = k\phi(n) + 1 \tag{1}$$

untuk suatu nilai k. Kita juga dapat memperluas Fungsi Totient Euler:

$$\phi(n) = (p-1)(q-1) = pq - p - q + 1,$$

karena p dan q jauh lebih kecil dibanding $pq = n$, dapat kita simpulkan bahwa,

$$\phi(n) \approx n$$

Jika kita membagi persamaan (1) dengan $d\phi(n)$, kita mendapatkan:

$$\frac{e}{\phi(n)} = \frac{k}{d} + \frac{1}{d\phi(n)}$$

Dengan menggunakan pendekatan $\phi(n) \approx n$ kita dapat menulis,

$$\frac{e}{n} \approx \frac{k}{d}$$

Jika kita mengetahui n dan $\phi(n)$, kita dapat dengan cepat memfaktorkan n. Pertimbangkan polinomial kuadrat:

$$(x-p)(x-q) = x^2 - (p+q)x + pq$$

Dengan substitusi variabel yang diketahui, kita dapat menulis:

$$x^2 - (n - \phi(n) + 1)x + n$$

Menggunakan rumus kuadrat, akar-akarnya dapat ditemukan dengan:

$$p, q = \frac{(n - \phi(n) + 1) \pm \sqrt{(n - \phi(n) + 1)^2 - 4n}}{2}$$

Jika solusi p dan q adalah bilangan bulat, maka serangan berhasil dan kunci privat d telah ditemukan.

III. Metode Penerapan Serangan Wiener

Evaluasi keamanan kriptografis memerlukan pemahaman mendalam tentang potensi kerentanan yang dapat dieksploitasi oleh pihak tidak bertanggung jawab. Dalam konteks Single Sign-On (SSO) Edunex yang mengintegrasikan ITB Account berbasis Microsoft Entra ID dengan Microsoft Authenticator, kriptografi asimetris RSA memiliki peran krusial, utamanya dalam mekanisme tanda tangan digital. Ketika Microsoft Entra ID mengeluarkan token identitas (misalnya, JSON Web Token dalam OpenID Connect atau SAML Assertion) kepada Edunex, token ini ditandatangani secara digital menggunakan kunci privat RSA milik IdP. Tanda tangan ini menjamin keaslian dan integritas token, memastikan bahwa token berasal dari sumber yang sah dan tidak dimodifikasi selama transmisi. Selain itu, RSA juga dapat digunakan untuk pertukaran kunci atau operasi kriptografi internal pada Identity Provider (IdP).

Metodologi ini dirancang untuk mendemonstrasikan secara konseptual Serangan Wiener terhadap implementasi RSA yang tidak tepat. Percobaan ini akan berfokus pada skenario hipotetis di mana kunci privat RSA yang digunakan memiliki eksponen privat d yang sangat kecil, membuatnya rentan terhadap serangan. Metodologi ini mencakup pembangkitan kunci RSA yang sengaja dibuat rentan, proses enkripsi pesan, dan penerapan Serangan Wiener untuk memulihkan kunci privat, diikuti dengan verifikasi dekripsi. Demonstrasi ini bertujuan untuk memberikan gambaran praktis tentang bagaimana Serangan Wiener bekerja dan implikasinya jika kondisi kerentanan terpenuhi dalam sistem yang menggunakan RSA.

A. `rsa_base.py`

1) Fungsi Utilisas Aritmetika Teori Bilangan:

- `is_prime(n)`: Mengecek apakah suatu bilangan prima.
- `gcd(a,b)`: Menghitung Pembagi Bersama Terbesar (PBB) menggunakan algorimta Euclidean.
- `extended_gcd(a,b)`: Algoritma Euclidean Diperluas untuk menemukan solusi persamaan $ax + by = gcd(a, b)$.
- `mod_inverse(a,m)`: Menghitung invers modular menggunakan hasil dari *extended* GCD.

2) Fungsi Utama:

- `generate_vulnerable_rsa_keys(p,q)`: Menghasilkan pasangan kunci RSA dengan kerentanan terkontrol, khususnya terhadap Serangan Wiener. Fungsi ini mengambil dua bilangan prima p dan q sebagai *input*, kemudian menghasilkan modulus N dan nilai $\phi(N)$ sebagai dasar perhitungan kunci. Pencarian eksponen privat d dimulai dengan mencari nilai dari 3 dan berlanjut secara berurutan hingga memenuhi syarat $d < \frac{1}{3}\sqrt[4]{N}$. Fungsi ini kemudian menghitung eksponen publik e sebagai invers modular dari d terhadap $\phi(N)$, sehingga membentuk pasangan kunci publik dan privat.

Algoritma Generasi Kunci RSA Rentan Wiener's Attack

```
def generate_vulnerable_rsa_keys(p, q):
    if not (is_prime(p) and is_prime(q)):
        raise ValueError("p dan q harus
bilangan prima")
    if p == q:
        raise ValueError("p dan q tidak
boleh sama")

    N = p * q
    phi_N = (p - 1) * (q - 1)
    wiener_bound = N**(0.25) / 3
    upper_bound = int(wiener_bound)

    if upper_bound < 3:
        raise Exception(f"Batas Wiener
terlalu kecil untuk p dan q ini")

    d = None
    for k in range(3, upper_bound + 1):
        if gcd(k, phi_N) == 1:
            d = k
            break

    if d is None:
        raise Exception("Tidak ditemukan
d yang valid")

    e = mod_inverse(d, phi_N)
    return (e, N), (d, N)
```

- `encrypt(message,public_key)`: Mengenkripsi pesan dengan rumus $ciphertext = message^e \bmod N$.
- `decrypt(ciphertext, private_key)`: Mendekripsi *ciphertext* dengan rumus $message = ciphertext^d \bmod N$.

Algoritma Fungsi Enkripsi dan Dekripsi

```
def encrypt(message, public_key):
    e, N = public_key
    return pow(message, e, N)

def decrypt(ciphertext, private_key):
    d, N = private_key
    return pow(ciphertext, d, N)
```

B. `wiener_demo.py`

- `continued_fraction_expansion(numerator, denominator)`: Menghitung representasi pecahan berlanjut dari suatu bilangan rasional $\frac{e}{N}$. Pecahan berlanjut adalah cara untuk merepresentasikan bilangan

real sebagai deret bilangan bulat. Berdasarkan landasan teori, serangan wiener memanfaatkan fakta bahwa jika d sangat kecil maka rasio $\frac{e}{N}$ akan menjadi aproksimasi yang sangat baik untuk $\frac{k}{d}$. Pecahan berlanjut yang dihasilkan akan menghasilkan konvergen dan salah satunya akan menjadi $\frac{k}{d}$ yang benar.

Algoritma Pecahan Berlanjut

```
def continued_fraction_expansion(
numerator, denominator):
    fractions = []
    while denominator != 0:
        quotient = numerator //
denominator
        fractions.append(quotient)
        numerator, denominator =
denominator, numerator % denominator
    return fractions
```

- `convergents_from_continued_fraction (fractions)`: Fungsi ini mengambil deret bilangan bulat yang dihasilkan oleh fungsi pecahan berlanjut dan menghitung "konvergen" dari pecahan berlanjut tersebut. Konvergen adalah aproksimasi rasional berturut-turut dari bilangan asli. Salah satu pasangan `(k, d_candidate)` akan menjadi kunci privat d yang sebenarnya.

Algoritma Mencari Konvergen dari Pecahan Berlanjut

```
def convergents_from_continued_fraction(
fractions):
    convergents = []
    n0, d0 = 0, 1
    n1, d1 = 1, 0

    for q in fractions:
        n = q * n1 + n0
        d = q * d1 + d0
        convergents.append((n, d))
        n0, d0 = n1, d1
        n1, d1 = n, d
    return convergents
```

- `wiener_attack(e, N)`: Fungsi ini bertujuan untuk memecahkan private key d pada RSA ketika nilainya terlalu kecil. Ini merupakan fungsi inti yang mengimplementasikan logika Serangan Wiener yang menerima e dan N sebagai *input*.

Algoritma Wiener's Attack

```
def wiener_attack(e, N):
    fractions =
continued_fraction_expansion(e, N)
    convergents =
convergents_from_continued_fraction(
fractions)

    for k, d_candidate in convergents:
        if d_candidate == 0:
            continue

        if k == 0:
            continue

        if (e * d_candidate - 1) % k ==
0:
            phi_N_candidate = (e *
d_candidate - 1) // k

            b_coeff = N - phi_N_candidate
 + 1
            discriminant = b_coeff**2 - 4
 * N

            if discriminant >= 0:
                sqrt_discriminant = int(
math.isqrt(discriminant))
                if sqrt_discriminant *
sqrt_discriminant == discriminant:
                    p_candidate = (
b_coeff + sqrt_discriminant) // 2
                    q_candidate = (
b_coeff - sqrt_discriminant) // 2

                    if p_candidate *
q_candidate == N and p_candidate > 1 and
q_candidate > 1:
                        return
d_candidate
    return None
```

Pertama, fungsi akan memanggil fungsi yang menghasilkan pecahan berlanjut. Kemudian, untuk mendapatkan semua konvergen fungsi akan memanggil fungsi yang menghasilkan konvergen. Lalu, iterasi dilakukan setiap `(k,d_candidate)` yang dihasilkan. Setiap pasangan akan masung ke pengecekan kondisi, pada kondisi pertama yaitu melewati kandidat d bernilai nol. Kemudian, terdapat kondisi k bernilai nol yang akan langsung dilewati untuk menghindari pembagian dengan nol. Masuk ke kondisi ketiga yang menjadi pemeriksaan krusial, yaitu jika `d_candidate` adalah d yang benar maka $(e \cdot d_{candidate} - 1$ harus habis dibagi k. Saat kondisi ini terpenuhi, fungsi akan menghitung kandidat untuk $\phi(N)$ lalu mencoba mencari p dan q. Jika p dan q berhasil ditemukan dan memenuhi dari syarat percobaan Serangan Wiener maka `d_candidate` yang diuji adalah d yang benar.

IV. Analisis dan Pembahasan

Percobaan Serangan Wiener yang dilakukan menunjukkan bagaimana RSA dapat menjadi titik kerentanan tunggal jika diimplementasikan secara tidak benar, khususnya pemilihan

parameter kunci yang lemah. Perlu diingat percobaan ini dilakukan secara konseptual dan teoritis, jadi tidak akan langsung menggambarkan keamanan sistem SSO Edunex. Berikut hasil demonstrasi Serangan Wiener yang berhasil dilakukan,

```
--- RSA dengan Kunci yang Rentan (untuk Demonstrasi Wiener's Attack) ---
p = 307, q = 353
N = 108371, phi(N) = 107712
Public Key (e, N) = (43085, 108371)
Private Key (d, N) = (5, 108371)
Kondisi kerentanan d < N^(1/4) / 3: 5 < 6.047933615005096 -> True
Pesan asli: 42
Ciphertext: 58287
Pesan didekripsi: 42
Dekripsi berhasil: True

--- Demonstrasi Wiener's Attack ---
Mencoba melancarkan Wiener's Attack dengan e = 43085, N = 108371
Serangan berhasil! Private exponent (d) yang ditemukan: 5
Private exponent (d) asli: 5
Apakah d yang ditemukan cocok dengan d asli? True
Pesan asli: 42
Pesan didekripsi menggunakan d yang ditemukan: 42
Dekripsi dengan d yang ditemukan berhasil: True
```

Gambar IV.1: Hasil dari percobaan Serangan Wiener dengan input $p = 307$ dan $q = 353$.

(Sumber: Arsip Penulis)

Berikut penjelasan secara matematis ketika $p = 307$ dan $q = 353$,

$$N = p \cdot q = 108.371,$$

$$\phi(N) = (307 - 1)(353 - 1) = 107.712,$$

Hitung batas Wiener,

$$\frac{1}{3}\sqrt[4]{N} = \frac{\sqrt[4]{108.371}}{3} \approx 6.047$$

Cari d mulai dari 3 yang relatif prima dengan $\phi(N)$, ditemukan saat $d = 5$ karena $gcd(5, 107712) = 1$ dan memenuhi syarat $5 < 6.047$. Kemudian lakukan perhitungan e,

$$e = d^{-1} \bmod \phi(N) = 5^{-1} \bmod 107712 = 43085,$$

(karena $5 \cdot 43085 = 215425 \equiv 1 \bmod 107712$).

Pecahan berlanjut $\frac{e}{N} = \frac{43085}{108371}$ akan menghasilkan konvergen yang mengandung $d = 5$ sehingga ditemukan d yang juga memenuhi syarat Wiener.

```
Terjadi kesalahan saat membuat kunci rentan: Tidak dapat menemukan
d yang sangat kecil (<3.366) dan coprime dengan phi_N.

--- Demonstrasi Wiener's Attack ---
Kunci rentan belum berhasil dibuat, tidak bisa melanjutkan serangan
```

Gambar IV.2: Hasil dari percobaan Serangan Wiener dengan input $p = 101$ dan $q = 103$.

(Sumber: Arsip Penulis)

Gambar IV.2 merupakan *output* yang diberikan ketika tidak dapat menentukan d yang sangat kecil hal ini akibat batas Wiener yang terlalu kecil. Berikut penjelasan secara matematsinya,

$$N = p \cdot q = 10.403,$$

$$\phi(N) = (101 - 1)(103 - 1) = 10.200,$$

Hitung batas Wiener,

$$\frac{1}{3}\sqrt[4]{N} = \frac{\sqrt[4]{10.403}}{3} \approx 3.366$$

Cari d mulai dari 3 hingga batas Wiener yang relatif prima dengan $\phi(N)$, hanya ada satu kemungkinan yaitu $d = 3$ tetapi karena $gcd(3, 10200) = 3$ sehingga tidak memenuhi syarat dan tidak ditemukan d valid yang bisa ditemukan. Hal ini karena rentang pencarian d yang terlalu sempit akibat batas Wiener yang terlalu kecil.

Secara hipotesis, kunci privat RSA yang digunakan Microsoft Entra ID untuk menandatangani token identitas memiliki eksponen privat d yang sangat kecil, maka Serangan Wiener secara teoritis dapat diterapkan. Penyerang yang berhasil mendapatkan kunci publik (N, e) dari sertifikat tanda tangan digital IdP dapat mencoba memulihkan d menggunakan teknik pecahan berlanjut. Jika penyerang berhasil memulihkan d, mereka dapat memalsukan tanda tangan digital pada token identitas. Hal ini memungkinkan mereka membuat token palsu yang sah dan mengakses akun SSO yang terhubung ke Edunex. Tentu ini berpotensi membahayakan data akademik dan informasi pribadi mahasiswa. Keberhasilan serangan ini juga akan secara masif merusak kepercayaan dan reputasi Edunex dan ITB.

Meskipun tidak berarti Edunex atau Microsoft Entra ID rentan, demonstrasi ini sebagai studi kasus untuk menekankan pentingnya pemilihan parameter kunci RSA yang kuat dan kepatuhan terhadap praktik terbaik kriptografi dalam sistem SSO apapun.

V. KESIMPULAN

Teori Bilangan merupakan fondasi matematis yang tak lepas dari keamanan kriptografi modern, khususnya algoritma RSA. Konsep-konsep seperti bilangan prima, Pembagi Bersama Terbesar (PBB), aritmetika modulo, fungsi totient Euler, dan algoritma Euclidean diperluas adalah kunci yang memungkinkan RSA berfungsi secara efektif.

Algoritma RSA, dengan proses pembangkitan kunci, enkripsi, dan dekripsi yang mengandalkan kesulitan memfaktorkan bilangan besar, telah menjadi standar keamanan digital. Namun, penelitian menunjukkan bahwa pemilihan parameter yang kurang tepat bisa membuka celah keamanan. Salah satunya adalah Serangan Wiener yang berhasil memecahkan RSA ketika eksponen privat d dipilih dengan nilai yang terlalu kecil hanya dengan memanfaatkan keunikan matematika pecahan berlanjut. Ini membuktikan bahwa meskipun RSA secara teori aman, implementasinya harus dibuat dengan sangat hati-hati. Serangan Wiener mengajarkan pentingnya menggunakan d yang cukup besar dan acak dalam pembangkitan kunci RSA.

Dalam konteks sistem SSO pada Edunex yang terintegrasi dengan ITB Account berbasis Microsoft, penggunan RSA paling mungkin terjadi pada proses tanda tangan digital token identitas. Jika kunci privat yang digunakan untuk tanda tangan ini memiliki d yang rentan, penyerang secara teoritis dapat

memalsukan token dan melakukan peniruan identitas. Namun, keberadaan Multi-Factor Authentication (MFA) melalui Microsoft Authenticator secara signifikan mengurangi dampak dari potensi kompromi kunci RSA, karena penyerang tetap harus mengatasi faktor otentikasi kedua. Intinya, meski ada celah teoretis di RSA, implementasi di ITB sudah relatif matang dengan perlindungan yang berlipat ganda. Namun, bukan berarti sudah sepenuhnya aman, melainkan keamanan siber itu proses terus menerus yang harus selalu dievaluasi.

VI. Lampiran

Berikut lampiran dari *source code* yang digunakan dalam demonstrasi *Wiener's Attack* yang mungkin bisa dikembangkan atau replika pada penelitian selanjutnya https://github.com/Rizelbit/Wiener-Attack-Makalah-Matdis.

Dalam mempermudah pemahaman mengenai makalah ini, video penjelasan mulai dari ide, konsep, dan proses demonstrasi telah disediakan yang dapat diakses pada tautan berikut https://youtu.be/NrAy02b1vKQ.

VII. Ucapan Terima Kasih

Pertama-tama, penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa karena berkat dan kasih karunia-Nya penulis diberikan kesempatan untuk menulis dan menyelesaikan makalah ini dengan baik dan semaksimal mungkin. Penulis juga mengucapkan terima kasih kepada keluarga yang terus mendukung dan memberikan semangat ketika sedang susah maupun senang. Tidak lupa juga penulis menyampaikan rasa terima kasih sebesar-besarnya kepada dosen pengampu mata kuliah IF1220 K-02 Arrival Dwi Sentosa, S.Kom., M.T. yang senantiasa berbagi ilmu dan mengajarkan banyak hal kepada penulis sebagai bekal untuk masa depan. Lebih lanjut lagi, terima kasih untuk teman-teman kelas K-02 yang telah memberikan segala dinamika di dalam maupun luar kelas. Terakhir, terima kasih kepada pribadi penulis yang mau terus belajar dan berjuang sehingga bisa menyelesaikan salah satu dari sekian banyaknya tugas dalam mencari kunci privat d yang bisa didekripsi pada masa yang akan datang.

Sastrawan berkebangsaan Rusia pernah berkata,

Dalam setiap titik yang kita tulis,
ada seluruh semesta
yang menunggu untuk ditemukan
dan ditinggalkan.

Fyodor Dostoevsky

Referensi

Pustaka

[1] R. Munir, "Teori Bilangan (Bag.1)," *Informatika STEI ITB*, Jun. 17, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/15-Teori-Bilangan-Bagian1-2024.pdf

[2] R. Munir, "Teori Bilangan (Bag.2)," *Informatika STEI ITB*, Jun. 17, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/16-Teori-Bilangan-Bagian2-2024.pdf

[3] R. Munir, "Teori Bilangan (Bag.3)," *Informatika STEI ITB*, Jun. 17, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/17-Teori-Bilangan-Bagian3-2024.pdf

[4] S. Wickramasinghe, "RSA algorithm in cryptography: Rivest Shamir Adleman explained," *Splunk Blog*, Jun. 18, 2025. [Online]. Available: https://www.splunk.com/en_us/blog/learn/rsa-algorithm-cryptography.html

[5] M. Anema, "RSA," Cornell University Mathematics Department, Jun. 18, 2025. [Online]. Available: https://pi.math.cornell.edu/~mec/2008-2009/Anema/numbertheory/rsa.html

[6] A. Kedari, "SAML vs OAuth/OpenID Connect: Selecting the right SSO protocol," *miniOrange Blog*, Jun. 18, 2025. [Online]. Available: https://www.miniorange.com/blog/saml-vs-oauth-comparison-for-atlassian-ecosystem/

[7] Frontegg, "What is SSO?," *Frontegg Guides*, Jun. 18, 2025. [Online]. Available: https://frontegg.com/guides/single-sign-on-sso

[8] Microsoft, "What is single sign-on? - Microsoft Entra ID," *Microsoft Learn*, Jun. 17, 2025. [Online]. Available: https://learn.microsoft.com/en-us/entra/identity/enterprise-apps/what-is-single-sign-on

[9] W. Susilo, J. Tonien, dan G. Yang, "The Wiener attack on RSA revisited: A quest for the exact bound," Singapore Management University, Jun. 19, 2025. [Online]. Available: https://ink.library.smu.edu.sg/cgi/viewcontent.cgi?params=/context/sis_research/article/8411/&path_info=The_Wiener_Attack_on_RSA_Revisited.pdf

[10] Cryptobook, "Wiener's attack," in *Cryptobook*, Jun. 19, 2025. [Online]. Available: https://cryptohack.gitbook.io/cryptobook/untitled/low-private-component-attacks/wieners-attack

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, 20 Juni 2025

Reinhard Alfonzo Hutabarat, 13524056