Implementasi Sistem Kriptografi Hybrid RSA dan DES untuk Pengamanan Data Teks

Putu Ayu Wulan Satya Dewi[a1], I Gede Santi Astawa[a2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]dewi.2308561008@student.unud.ac.id
[2]santi.astawa@unud.ac.id

Abstract

Data security is a crucial aspect in the digital age. Symmetric algorithms such as the Data Encryption Standard (DES) offer high-speed encryption, but have fundamental weaknesses in terms of secure key distribution. On the other hand, asymmetric cryptography such as RSA can overcome key distribution issues, but is computationally inefficient for encrypting large volumes of data. To address these challenges, this article demonstrates the implementation of a hybrid cryptographic system that integrates the strengths of both algorithms. This system leverages the RSA algorithm, which is based on principles of number theory such as modular arithmetic and the difficulty of prime factorization, to securely encrypt and distribute DES session keys. Subsequently, the DES algorithm is used to quickly and efficiently encrypt textual data. Through an explanation of the mathematical processes and programmed implementation, this research demonstrates that the hybrid approach can both secure and restore plaintext data intact.

Keywords: *Hybrid Cryptography, RSA, DES, Key Exchange, Number Theory*

1. Pendahuluan

Pertukaran data menjadi bagian yang tidak terpisahkan pada era digital saat ini. Kejahatan siber seperti modifikasi data mengancam kerahasiaan, *data integrity*, *authentication* dan *non repudiation* [1]. Oleh karena itu, cara untuk menjaga keamanan pesan adalah dengan kriptografi. Kriptografi adalah ilmu yang digunakan untuk menggubah pesan menjadi bentuk yang tidak bermakna agar tidak dapat dibaca oeh pihak yang tidak berwenang, sehingga pesan terjaga keamanannya. Berdasarkan jenis kuncinya, algoritma kriptografi dibagi menjadi dua jenis yaitu simetris dan asimetris. Algoritma kriptografi simetris hanya menggunakan satu kunci untuk proses enkripsi dan dekripsi pesan. Sedangkan, algoritma kriptografi asimetris menggunakan sepasang kunci yang terdiri dari kunci publik dan kunci privat [2].

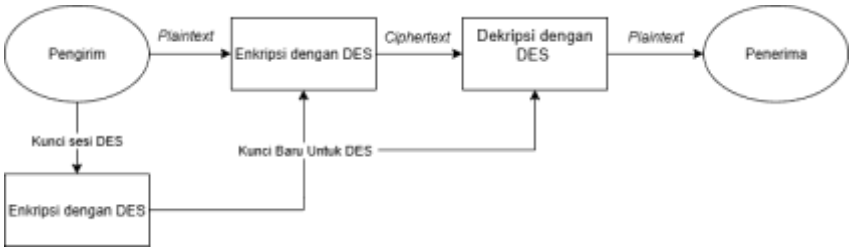
Algoritma kriptografi asimetris dan simetris, masing-masing memiliki kekurangan. Kriptografi simetris unggul dalam kecepatan namun lemah dalam distribusi kunci. Sementara kriptografi asimetris unggul dalam distribusi kunci namun lambat dalam waktu komputasi [3]. Maka dari itu, penggabungan algoritma simetris dan asimetris dikombinasikan untuk saling menutupi kekurangan. Penggabunggan ini dikenal dengan istilah kriptografi *hybrid* [4], di mana algoritma asimetris seperti RSA digunakan untuk mengenkripsi kunci DES, sedangkan algoritma simetris seperti DES digunakan untuk mengenkripsi isi pesan.

Dengan menggabungkan keunggulan kedua algoritma tersebut, sistem kriptografi *hybrid* mampu meningkatkan keamanan dan performa dalam pertukaran data teks. Oleh karena itu, penelitian ini membahas penerapan teori bilangan dalam perancangan sistem kriptografi *hybrid* RSA-DES sebagai solusi untuk menjaga kerahasiaan dan integritas data secara efisien dan aman.

2. Metode Penelitian

2.1 Perancangan Arsitektur Sistem

Sistem dirancang dengan mengkombinasikan keunggulan dari masing-masing algoritma DES dan RSA. Arsitektur sistem ini bertujuan untuk memanfaatkan kecepatan enkripsi DES untuk data yang berukuran besar dan keamanan distribusi kunci dari RSA. Alur kerja sistem secara umum diilustrasikan pada Gambar 1.

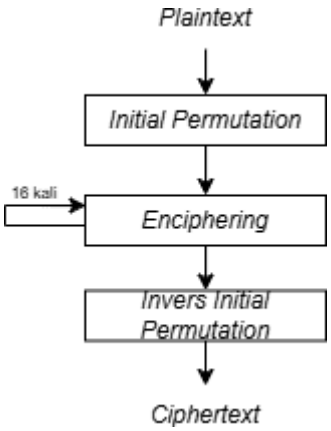


Gambar 1. Proses Algoritma Hybrid DES dan RSA

Proses dimulai di sisi pengirim, di mana sebuah kunci sesi DES acak dibangkitkan. Kunci sesi ini digunakan untuk mengenkripsi pesan (*plaintext*) menjadi *ciphertext*. Kunci sesi itu sendiri dienkripsi menggunakan kunci public RSA milik penerima. Kedua hasil enkripsi, yaitu *ciphertext* pesan dan *ciphertext* kunci sesi, kemudian dikirimkan ke penerima. Penerima pertama-tama menggunakan kunci privat RSA miliknya untuk mendekripsi ciphertext kunci sesi. Setelah kunci sesi DES asli berhasil didapatkan, kunci tersebut digunakan untuk mendekripsi *ciphertext* pesan menjadi *plaintext* semula.

2.2 Skema Algoritma DES

DES merupakan kriptografi simetris yang tergolong dalam *cipher blok* yang beroperasi pada blok berukuran 64 bit [5]. *Plaintext* berukuran 64 bit dienkripsi menjadi 64 bit *ciphertext* menggunakan 56 bit *internal key* yang dibangkitkan dari kunci eksternal.



Gambar 2. Skema Global DES

Proses enkripsi DES diawali dengan *initial permutation* terhadap *block plaintext* yang diacak urutan bitnya. Setelah itu, *block* tersebut di *enciphering* sebanyak 16 kali. Sebagai langkah terakhir, hasil dari 16 putaran tersebut dipermutasi dengan *invers initial permutation* menjadi *block ciphertext*.

2.3 Skema Algoritma RSA

Algoritma RSA merupakan algoritma kunci publik yang berjenis enkripsi asimetris menggunakan dua kunci yang berbeda yaitu kunci publik dan kunci privat[6]. Untuk membangkitkan sepasang kunci, dilakukan dengan langkah-langkah berikut:

- Menentukan dua bilangan prima p dan q secara sembarang dan tidak sama.
- Menghitung modulus(n) dengan p x q.
- Menghitung nilai dari f(n) = (p– 1) (q– 1)
- Memilih kunci public e yang bersifat relatif prima dengan hasil nilai f(n).
- Untuk pembangkitan kunci privat dihitung menggunakan persamaan ed □ 1 (mod □(n)) atau d□ e^{-1} mod (□(n)).

Pembangkitan sepasang kunci tersebut menghasilkan kunci publik (e, n) dan kunci privat (d,n). Proses enkripsi menggunakan rumus $c_i = m_i^e \bmod n$. Untuk proses dekripsi menggunakan rumus $m_i = c_i^d \bmod n$.

3. Hasil dan Diskusi

Sebagai ilustrasi, misalkan kita memiliki sebuah pesan dan ingin mengamankannya dengan menggunakan sistem kriptografi *hybrid*. Dalam sistem ini, algoritma DES digunkan untuk mengenkripsi pesan, sedangkan algoritma RSA untuk mengenkripsi kunci sesi DES. Adapun isi pesan yang akan dienkripsi sebagai berikut.

Pesan = “SEMANGAT SNATIA”

3.1. Proses Enkripsi Pesan Secara Matematis

Proses enkripsi dilakukan oleh pengirim

a. Konversi Pesan ke Biner
Langkah pertama dalam proses enkripsi adalah mengkonversi setiap karakter pada pesan “SEMANGAT SNATIA” menjadi nilai ASCII, lalu ke biner. Panjang pesan sebanyak 14 karakter diproses per 8 byte dalam blok.

Tabel 1. Konversi Pesan ke ASCII dan Biner

Letter	ASCII	Biner
S	83	01010011
E	69	01000101
M	77	01001101
A	65	01000001
N	78	01001110
G	71	01000111
A	65	01000001
T	84	01010100
S	83	01010011
N	78	01001110
A	65	01000001
T	84	01010100

Letter	ASCII	Biner
I	73	01001001
A	65	01000001

b. Initial Permutation (IP)

Setelah pesan dikonversi ke dalam bentuk biner, data 64 bit akan di IP berdasarkan tabel standar DES. Tahap ini untuk mengubah urutan bit-bit input sebagai persiapan untuk langkah selanjutnya.

c. Struktur Feistel 16 Putaran

Setelah Initial Permutation, data akan melalui 16 putaran dari struktur Feistel. Untuk setiap putaran. Untuk setiap putaran i = 1 hingga 16 berlaku persamaan matematis sebagai berikut:

L_i = R_{i-1}
R_i = L_{i-1} $\oplus$ f(R_{i-1}, K_i)

$\oplus$ = XOR
f() = fungsi substitusi dan permutasi (melibatkan S-box dan E-box). K_i = sub-kunci dari K_DES yang dihasilkan melalui key schedule.

d. *Final Permutation*

Setelah melalui 16 putaran Feistel, hasil dari putaran akhir (L_16 dan R_16) akan digabungkan kembali.Susunan bit dari blok yang sudah menyatu ini kemudian diacak untuk terakhir kalinya. *Output* dari tahap ini adalah *block ciphertext* 64 bit yang terenkripsi oleh DES.

3.2. Proses Enkripsi Kunci Sesi DES dengan RSA

Setelah mengenkripsi pesan, sistem *hybrid* ini mengenkripsi kunci sesi DES menggunakan algoritma RSA. Untuk tujuan ilustrasi, menggunakan parameter kunci publik RSA sebagai berikut:

Nilai e = 65537
N = 3233 (merupakan hasil perkalian p x q)

a. Representasi Kunci DES Sebagai Integer

Kunci DES (K_DES) 0xA1B2C3D4E5F60708 yang berformat heksadesimal harus direpresentasikan sebagai bilangan bulat desimal agar dapat dioperasikan secara matematis dengan RSA. K_DES dalam desimal = 11670881217876308552.

b. Enkripsi RSA

Kunci DES yang telah dikonversi ke integer kemudian dienkripsi menggunakan rumus dasar RSA. $C_{RSA} = K\_DES_{int}^{e} \pmod n$

Hasil dari perhitungan ini adalah kunci DES yang telah dienkripsi dengan RSA yang siap untuk dikirimkan bersama *ciphertext.*

3.3. Proses Dekripsi

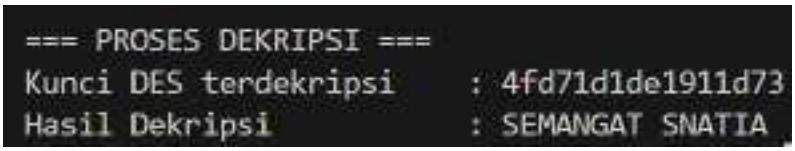
Di sisi penerima, proses dekripsi dilakukan untuk memulihkan *plaintext*. Hasil proses ini dapat dilihat pada Gambar 5. Penerima menggunakan kunci privat RSA miliknya untuk mendekripsi kunci DES yang telah dienkripsi RSA. Proses dekripsi dengan PKCS1_OAEP berhasil untuk mengembalikan kunci sesi DES yang asli. Setelah kunci sesi DES berhasil dipulihkan, kunci digunakan untuk mendekripsi *ciphertext* DES.

3.4. Implementasi Sistem

Sistem kriptografi hybrid ini diimplementasikan menggunakan bahasa pemrograman Python dengan bantuan library PyCryptodome. Dalam implementasi, fungsi enkripsi pertama-tama, membangkitkan kunci sesi DES 64-bit yang unik menggunakan get_random_bytes. Selanjutnya, pesan teks diproses dengan fungsi pad() untuk memastikan ukurannya sesuai dengan kelipatan blok DES, sebelum dienkripsi menggunakan objek cipher DES.new() dalam mode ECB. Untuk mengamankan kunci sesi, sistem membangkitkan sepasang kunci RSA dan menggunakan kunci publiknya untuk mengenkripsi kunci DES melalui skema PKCS1_OAEP. Proses dekripsinya dimulai dengan menggunakan kunci privat RSA untuk mendekripsi kunci sesi DES yang diterima, yang secara efektif membalikkan proses enkripsi PKCS1_OAEP. Setelah kunci sesi asli berhasil dipulihkan, kunci tersebut digunakan untuk menginisialisasi objek cipher DES kedua, yang kemudian mendekripsi ciphertext utama dan menghilangkan *padding* dengan fungsi unpad() untuk mengembalikan *plaintext* ke bentuk semula. Berikut adalah contoh hasil eksekusi program:

Gambar 3. Input untuk Proses Enkripsi

Gambar 4. Hasil Enkripsi Eksekusi Program



Gambar 5. Hasil Dekripsi Eksekusi Program

4. Kesimpulan

Penelitian ini, berhasil merancang dan mengimplementasikan sistem kriptografi *hybrid* yang mengintegrasikan algoritma RSA dan DES untuk pengamanan data teks. Melalui implementasi menggunakan bahasa pemrograman Python dan library PyCryptodome, sistem ini terbukti mampu melakukan proses enkripsi dan dekripsi secara fungsional. Pesan teks dienkripsi menggunakan DES, sementara kunci sesi DES yang diamankan menggunakan enkripsi asimetris RSA. Hasil penelitian menunjukkan bahwa pendekatan hybrid secara efektif menggabungkan kecepatan algoritma simetris dengan keamanan distribusi kunci dari algoritma asimetris, menghasilkan solusi yang baik antara performa dan keamanan.

Daftar Pustaka

[1] R. Munir, “01 - Pengantar Kriptografi.” Accessed: Jun. 28, 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2022-2023/01-Pengantar-Kriptografi-(2023).pdf

[2] A. Widyanto, “Implementasi Kriptografi Teks Menggunakan RSA,” *Community Serv. Artic. e-ISSN*, vol. 1, no. 2, pp. 70–81, 2024.

[3] H. Putri, L. Virna, T. Febrianti, and T. Sutabri, “Pengamanan Data Transmisi Aplikasi Web Menggunakan Algoritma Kriptografi RSA: Studi Kasus dan Analisis,” vol. 5, no. 1, pp.

153–170, 2025.
[4] H. Andara, F. Damayanti, and Khairunnisa, “Implementasi Kriptografi Hybrida Algoritma RSA dan Vernam Cipher Dalam Pengamanan File Text,” *J. Ilmu Komput. dan Inform.*, vol. 6, no. 1, pp. 8–15, 2022.
[5] A. Ariska and W. Wahyuddin, “Penerapan Kriptografi Menggunakan Algoritma Des (Data Encryption Standard),” *J. Sintaks Log.*, vol. 2, no. 2, pp. 9–19, 2022, doi: 10.31850/jsilog.v2i2.1734.
[6] R. Munir, “Algoritma RSA Bahan kuliah IF4020 Kriptografi,” 2020, [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Algoritma-RSA-2020.pdf

.