JURNAL SITEBA
Volume 2, No. 2, 2023
https://journal.iteba.ac.id/index.php/jurnalsiteba/index

ANALISIS PERBANDINGAN TINGKAT KOMPLEKSITAS WAKTU ENKRIPSI DAN TINGKAT KEAMANAN ENKRIPSI PADA ALGORITMA KRIPTOGRAFI RSA, DES, AES

Muhammad Zulfikar[1], Tapha Imanuddin[2], Nova Eko Prastyo[3]
Satriya Adi Firmansyah[4], Riyan Anugrah Alhad[5]
zulfikar@iteba.ac.id

Abstract

In today's era of advanced information and technology, data plays an important role. However, the importance of data also brings challenges in terms of security and privacy. Cryptography is an effective solution to maintain data security by encoding information so that only authorized parties can read it. This research aims to compare cryptography algorithms such as RSA, DES, AES. This comparison will give an idea about the security level of each algorithm in various situations. RSA is a strong public key cryptography algorithm, using large number factorization for its security. RSA encryption requires O(n^k) time complexity, where n is the number of bits in the public key and k corresponds to the factorization algorithm. The longer the key, the higher the security level, but the encryption process can be slower. DES is a symmetric key cryptography algorithm with 64-bit blocks and O(1) time complexity in the encryption process. However, the relatively short key length makes DES vulnerable to brute-force attacks. AES is a symmetric key cryptography algorithm with 128-bit blocks and 128-bit keys. The AES encryption process has O(1) time complexity, offers better security with longer key sizes and complex structures, and reduces the risk of brute-force attacks. AES is a highly secure and efficient encryption algorithm for maintaining data confidentiality and integrity.

1. PENDAHULUAN

Data memiliki peranan penting dalam era informasi dan teknologi yang maju saat ini. Namun, pentingnya data juga membawa tantangan dalam hal keamanan dan privasi.

Kriptografi menjadi solusi efektif untuk menjaga keamanan data dengan menyandikan informasi sehingga hanya pihak berwenang yang dapat membacanya. Namun, beberapa algoritma kriptografi yang dahulu dianggap kuat dan aman, kini terbukti lemah terhadap serangan komputasi canggih. Selain itu, penyimpanan kunci yang buruk juga dapat menyebabkan masalah keamanan pada sistem kriptografi. Penelitian ini bertujuan membandingkan algoritma kriptografi seperti RSA, DES, AES. Perbandingan ini akan memberikan gambaran tentang tingkat keamanan masing-masing algoritma dalam berbagai situasi. Manajemen kunci yang baik juga menjadi sorotan untuk memastikan tingkat keamanan yang maksimal dalam implementasi algoritma kriptografi. Keamanan data merupakan isu yang terus berkembang seiring perkembangan teknologi. Penelitian ini diharapkan memberikan pemahaman lebih lanjut tentang kriptografi dan membantu memilih algoritma yang tepat dalam konteks keamanan data (1)

A. Algoritma RSA (Rivest Shamir Adleman)

Penerapan algoritma RSA untuk enkripsi dan dekripsi teks dengan cara mengubah setiap huruf di dalam teks menjadi angka sesuai yang ditentukan, kemudian menentukan p dan q bernilai dua bilangan prima besar, acak dan dirahasiakan, $p \neq q$ setelah itu menhitung nilai $n = p \times q$, dan hitung $\phi(n) = (p - 1) \times (q - 1)$, nilai n disebut (RSA) modulus, untuk menentukan nilai e dengan cara menentukan bilangan prima acak yang memiliki syarat $1 < e < \phi(n)$ (2). Untuk menentukan nilai d dapat dilakukan dengan persamaan $(d * e) \bmod \varphi(n) = 1$ (3). Untuk melakukan proses enkripsi maka dilakukan perhitungan dengan menggunakan pasangan kunci yang sudah dibangkitkan dan rumus enkripsi algoritma RSA berikut. $C_i = P_i^e \bmod n$ (4)

B. Algoritma DES

Algoritma kriptografi simetris DES menggunakan kunci 56 bit untuk mengenkripsi dan mendekripsi blok data 64 bit. DES terdiri dari tiga proses: pembangkitan kunci internal, enkripsi data, dan dekripsi data. Kunci eksternal (64 bit) dimasukkan oleh pengguna, lalu diubah menjadi kunci internal (56 bit) untuk melakukan 16 putaran enkripsi. Proses dekripsi menggunakan kunci yang sama dengan kunci enkripsi. DES mengubah 64 bit plaintext menjadi 64 bit ciphertext dalam proses enkripsi. Meskipun DES telah banyak digunakan, namun kunci 56 bitnya kini dianggap kurang aman karena kemajuan teknologi komputasi. Oleh karena itu, algoritma dengan kunci lebih panjang disarankan untuk tingkat keamanan yang lebih tinggi.(5)

C. Algoritma AES (Advanced Encryption Standart)

Algoritma kriptografi simetris AES digunakan secara luas dan diadopsi sebagai standar enkripsi oleh banyak negara, termasuk Amerika Serikat. AES menggunakan kunci dengan panjang 128, 192, atau 256 bit untuk mengenkripsi data dalam 32 putaran dengan 128-bit subkunci. Ini adalah algoritma yang sangat kuat dan belum ada laporan serangan berhasil merusaknya melalui kriptanalisis.

Namun, AES memiliki kelemahan, terutama pada versi AES-256, yang dapat dianalisis menggunakan serangan Rectangle Algebraic dengan 10 putaran. Meskipun demikian, jumlah kunci yang perlu ditebak untuk serangan ini sangat besar (2131,8 kelompok kunci), sehingga tetap sulit untuk dilakukan secara praktis.

2. METODE

Dalam penelitian ini, kami akan melakukan analisis perbandingan beberapa algoritma kriptografi, yaitu RSA, DES, dan AES, meliputi pengumpulan data, analisis kecepatan enkripsi, analisis keamaman enkripsi.

2.1 Pengumpulan Data

Pengumpulan data berasal dari studi literatur terkait masing masing algoritma.

2.2 Analisis Kecepatan Enkripsi Data

Analisis kecepatan algoritma dalam enkripsi data dilakukan dengan menghitung jumlah kunci, kompleksitas algoritma, metode enkripsi yang digunakan (algoritma bekerja pada blok data atau mengalir), dan ukuran data yang ingin dienkripsi.

2.3 Analisis Keamanan Enkripsi Data

Analisis keamaan algoritma didapatkan dari panjang kunci algoritma, cara algoritma menghasilkan kunci dan mendistribusikan kunci, penanganan kesalahan, dan perlindungan kunci.

3. HASIL DAN PEMBAHASAN

3.1 Algoritma RSA

Pilih bilangan prima acak p dan q:

p = 153738759935799436O9

q = 14587168563558136397

Hitung nilai n = p * q:

n = 22429307220748206192438268537247459287 3

Hitung nilai phi(n) = (p - 1) * (q - 1):

phi(n) = 22429307220748206186729731898093758796 8

Pilih bilangan bulat e, 1 < e < phi(n), dan relatif prima dengan phi(n):

e = 65537

Hitung nilai d = e^(-1) mod phi(n):

d = 16163583204870841237812341460101163014 5

Kunci publik: (e, n) -> (65537,

22429307220748206192438268537247459287 3)

Kunci privat: (d, n) -> (16163583204870841237812341460101163014 5,

22429307220748206192438268537247459287 3)

untuk melakukan enkripsi data harus diubah menjadi angka dan dilakukan perhitungan dengan rumus C = M^e mod n, contoh:

Enkripsi untuk huruf N yang diubah ke bilangan ASCI:

C(N) = N^e mod n

C(N) = 78^65537 mod 22429307220748206192438268537247459287 3

C(N) = 18305762422935732204785954161034749502 3

3.2 Alogritma DES

Untuk melakukan enkripsi dengan Algoritma DES, data yang ingin di enkripsi harus diubah dahulu menjadi data biner yang sudah melalui proses pemutasi dengan panjang 64 bit dan menentukan kunci berupa data biner dengan panjang 64 bit, contoh penerapan algoritma DES:

Langkah 1: Inisialisasi

M = 1111000010101010111100000110010101010001000000100001101 1

K=1010101010101010101010101010101010101010101010101010101010101010

Langkah 2: Pembagian menjadi Blok

Data hasil permutasi awal (M) akan dibagi menjadi dua bagian, masing-masing bagian memiliki panjang 32 bit. Bagian pertama disebut Left (L0) dan bagian kedua disebut Right (R0).

L0 = 1111000010101010111100000110010 1

R0 = 0101000100000010000011011

Langkah 3: 16 Iterasi Proses Feistel

Proses DES melibatkan 16 iterasi, di mana pada setiap iterasi dilakukan ekspansi, fungsi F, dan permutasi.

Iterasi 1:

L1 = R0 = 01010001000000100000011011

R1 = L0 xor F(R0, K1)

Iterasi 2:

L2 = R1 = 10010000001000000100011001

R2 = L1 xor F(R1, K2)

...

Iterasi 16:

L16 = R15 = 10000001011011100010101l0

R16 = L15 xor F(R15, K16)

Langkah 4: Penukaran dan Penggabungan

Setelah 16 iterasi selesai, maka nilai L16 dan R16 akan ditukar, dan kemudian digabungkan.

R16L16 = 10000001011011100010101101001000000010010000l010

Langkah 5: Permutasi Akhir

Data R16L16 akan dilakukan permutasi akhir (final permutation) untuk menghasilkan ciphertext (C).

C = 01010000001001111000101010100101000101010000010O

Sehingga, hasil enkripsi DES dari data biner M dengan kunci K adalah:

C = 01010000001001111000101010100101000101010000010O

3.3 Algoritma AES

Untuk melakukan enkripsi dengan algoritma AES data harus kita ubah dulu ke dalam bentuk hexadesimal dengan panjang 128 bit dan menentukan kunci dalam hexadesimal dengan panjang 128 bit juga, contoh penerapan:.

data = 4E 61 6D 61 20 73 61 79 61 20 6D 75 68 61 6D 61

kunci = 6B 75 6E 63 69 72 61 68 61 73 69 61 31 32 33 34

lakukan operasi XOR dengan kunci: 4E 61 6D 61 20 73 61 79 61 20 6D 75 68 61 6D 61 XOR 6B 75 6E 63 69 72 61 68 61 73 69 61 31 32 33 34 = "5D 24 4A E2 A9 44 C5 8B 9D 2B A2 17 C9 1E B8 63"

Lakukan 9 ronde (rounds) pertama:

SubBytes: Terapkan substitusi S-box pada setiap byte dalam blok.

ShiftRows: Geser baris pada setiap kolom.

MixColumns: Campur kolom dalam blok.

AddRoundKey: Tambahkan kunci putaran.

Lakukan ronde ke-10:

SubBytes

ShiftRows

AddRoundKey

Langkah 5: Hasil Enkripsi

Hasil dari enkripsi adalah blok teks yang terenkripsi dalam bentuk HEX. Contoh hasil:

Hasil: "18 51 27 C7 4F 66 FC 20 37 95 1E 15 A9 C3 D0 33 3D 06 48 9C 6D FC 6A D1 D2 F0 44 4A 71 77 A1 52 68 85 91 E1 5D 2B 0C 99 D9 E8 B1 E6 79 8F 92 B5 C7 6D 68 0B"

3.4 Hasil perbandingan setiap algoritma

a) RSA (Rivest-Shamir-Adleman):

i. RSA adalah algoritma kriptografi kunci publik yang menggunakan konsep faktorisasi bilangan besar untuk keamanannya.

ii. Kompleksitas waktu pada proses enkripsi RSA adalah O(n^k), di mana n adalah jumlah bit dalam kunci publik dan k adalah bilangan yang berkaitan dengan algoritma faktorisasi yang digunakan (biasanya sekitar 2 hingga 3) .

iii. Kompleksitas enkripsi RSA dapat membuat proses enkripsi menjadi lebih lambat dengan kunci yang lebih panjang untuk meningkatkan tingkat keamanan.

b) DES (Data Encryption Standard):

i. DES adalah algoritma kriptografi kunci simetris yang menggunakan blok 64-bit.

ii. Kompleksitas waktu pada proses enkripsi DES adalah O(1), karena langkah-langkah enkripsi yang dilakukan memiliki jumlah tetap dan tidak bergantung pada ukuran data yang dienkripsi.

iii. panjang kunci DES yang relatif pendek, algoritma ini rentan terhadap serangan brute-force menggunakan komputasi modern.

c) AES (Advanced Encryption Standard):

i. AES merupakan algoritma kriptografi kunci simetris yang menggunakan blok 128-bit dan kunci 128-bit.
ii. Kompleksitas waktu pada proses enkripsi AES dapat dianggap sebagai O(1) karena langkah-langkah enkripsi yang dilakukan memiliki jumlah tetap.
iii. AES menawarkan keamanan yang lebih baik karena ukuran kunci yang lebih panjang dan struktur yang lebih kompleks, sehingga mengurangi kemungkinan serangan brute-force.

4. KESIMPULAN

RSA adalah algoritma kriptografi kunci publik yang kuat, menggunakan faktorisasi bilangan besar untuk keamanannya. Proses enkripsi RSA memerlukan kompleksitas waktu O(n^k), dimana n adalah jumlah bit dalam kunci publik dan k berkaitan dengan algoritma faktorisasi. Semakin panjang kunci, semakin tinggi tingkat keamanannya, tetapi proses enkripsi dapat menjadi lebih lambat. DES adalah algoritma kriptografi kunci simetris dengan blok 64-bit dan kompleksitas waktu O(1) dalam proses enkripsi. Namun, panjang kunci yang relatif pendek membuat DES rentan terhadap serangan brute-force. AES adalah algoritma kriptografi kunci simetris dengan blok 128-bit dan kunci 128-bit. Proses enkripsi AES memiliki kompleksitas waktu O(1), menawarkan keamanan yang lebih baik dengan ukuran kunci yang lebih panjang dan struktur yang kompleks, serta mengurangi risiko serangan brute-force. AES adalah algoritma enkripsi yang sangat aman dan efisien untuk menjaga kerahasiaan dan integritas data.

REFERENSI

1. MASRIL MA, CANIAGO DP. Sistem Pencegahan Illegal Fishing di Laut Batam menggunakan YOLOv7 berbasis Notifikasi Telegram. ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika [Internet]. 2024 Jan 17;12(1):175. Available from: https://ejurnal.itenas.ac.id/index.php/elkomika/article/view/10020
2. Gunawan I. Kombinasi Algoritma Caesar Cipher dan Algoritma RSA untuk pengamanan File Dokumen dan Pesan Teks. InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan). 2018;2(2):124–9.
3. Prasetyo Y, Triandi B, Hardianto H. Perancangan Aplikasi Pengamanan File Teks dengan Skema Hybrid Menggunakan Algoritma Enigma dan Algoritma RSA. It (Informatic Technique) Journal. 2018;6(1):46.
4. Rambe MR, Haryanto EV, Setiawan A. Aplikasi Pengamanan Data Dan Disisipkan Pada Gambar Dengan Algoritma Rsa Dan Modified Lsb Berbasis Android. It (Informatic Technique) Journal. 2019;7(1):51.
5. Suparman B. Aplikasi Pengamanan Data Menggunakan Kriptografi Dengan Metode Wake dan Algoritma Des Bebasis Java Desktop. OKTAL: Jurnal Ilmu Komputer dan Sains. 2022;1(07):808–17.