JNATIA Volume 1, Nomor 1, November 2022
Jurnal Nasional Teknologi Informasi dan Aplikasinya

Penerapan Algoritma Advanced Encryption Standard (AES- 128) Dengan Mode ECB Dalam Pengamanan File

Haposan Simangunsong[1], Made Agung Raharja[2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
sanphosansan@gmail.com[1] made.agung@unud.ac.id[2]

Abstract

Data Protection is one of the important things to protect, important messages and information from corruption, compromise or loss so that messages and information remain safe. Encryption and decryption techniques can secure data properly by protecting files from being easily read or seen by unauthorized parties. In this case, the authors used a cryptography symmetric algorithm called Advanced Encryption Standard (AES) and Electronic Code Book (ECB) as a solution to existing problems. The AES algorithm process is divided into four steps, the first step is Sub Bytes, the second step is Shift Rows, the third step is Mix Columns, and the las step is Add Round Key. And using the SHA algorithm as the hashing function. The algorithm is applied to a web application and using python as a programming language. Advanced Encryption Standard (AES) Algorithm with ECB Mode can be implemented to encrypt and decrypt files. The use of AES will encrypt every 128bit block of the file until it becomes a ciphertext which is an array of encrypted bytes. The decryption process using the AES algorithm with ECB mode will decrypt every 128 bits of the ciphertext to produce the original byte array file. AES with ECB mode which is implemented in the python programming language can be used to encrypt media files such as images, audio and video with a good level of security.

Keywords: *Data Protection, Encryption, Decryption, Algorithm AES-128, ECB*

1. Introduction

Data pribadi sering menjadi sasaran pihak yang tidak bertanggung jawab untuk dimanipulasi atau disalahgunakan. Oleh karena itu, penting untuk menjaga keamanan data pribadi atau rahasia. Salah satu teknik yang dapat digunakan untuk mengamankan data adalah enkripsi. Enkripsi adalah proses penyandian pesan atau informasi yang semula dapat dimengerti menjadi bentuk yang sulit atau bahkan tidak dapat dimengerti. Teknik ini melindungi data dengan mengubahnya menjadi format yang tidak terbaca tanpa kunci yang tepat. Setelah data dienkripsi, proses dekripsi dengan kunci yang sesuai dapat mengembalikan data ke bentuk aslinya. Enkripsi meningkatkan keamanan data yang bersifat sensitif, seperti file penting atau rahasia. Kriptografi, yang merupakan cabang ilmu matematika, berfokus pada aspek keamanan informasi, seperti integritas data dan keaslian entitas serta data itu sendiri. Dalam kriptografi, terdapat dua proses utama, yaitu enkripsi dan dekripsi. Enkripsi mengubah data asli menjadi bentuk yang tidak terbaca, sementara dekripsi mengubahnya kembali ke bentuk yang dapat dimengerti. Salah satu ancaman keamanan yang mengharuskan penggunaan enkripsi adalah ransomware, yang menyerang sistem komputer untuk mengenkripsi file di dalamnya, seperti yang terjadi pada rumah sakit di Amerika Serikat pada 29 September 2020. Untuk memitigasi risiko tersebut, enkripsi file sangat diperlukan. Salah satu metode kriptografi yang digunakan untuk pengamanan data adalah Advanced Encryption Standard (AES), yang bekerja dengan metode block cipher, memproses data dalam blok 128-bit dengan kunci 128-bit, 192-bit, atau 256-bit. Salah satu mode pada AES adalah Electronic Code Book (ECB), yang sering digunakan untuk enkripsi data.

2. Reseach Methods

2.1. AES

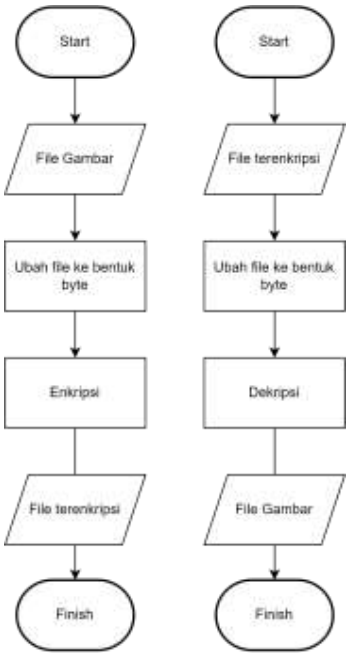
AES adalah algoritma enkripsi berbasis jaringan substitusi-permutasi yang efisien pada perangkat lunak dan keras, berbeda dengan DES karena tidak menggunakan jaringan Feistel. AES adalah variasi dari Rijndael dengan ukuran blok 128-bit dan ukuran kunci 128, 192, atau 256-bit. Proses kerja AES meliputi: pertama, ekspansi kunci untuk menghasilkan kunci ronde 128-bit; kemudian, penambahan kunci ronde awal dengan operasi XOR. Selama 9, 11, atau 13 ronde, dilakukan SubBytes (substitusi non-linear), ShiftRows (pergeseran byte), MixColumns (pencampuran byte), dan AddRoundKey (penambahan kunci ronde). Pada ronde terakhir, hanya SubBytes, ShiftRows, dan AddRoundKey yang diterapkan tanpa MixColumns.

2.2. ECB

ECB (Electronic Code Book) adalah mode block cipher yang paling sederhana, di mana setiap blok input plaintext dienkripsi langsung menjadi blok ciphertext. Pesan yang lebih besar dari ukuran blok dapat dibagi menjadi beberapa blok dan enkripsi dilakukan secara berulang. Keuntungan ECB adalah memungkinkan enkripsi paralel pada blok-bit, sehingga lebih cepat.

2.3. Gambaran Umum Sistem

Gambaran dari sistem yang dibangun untuk melakukan enkripsi dan dekripsi dari *file* dapat dilihat pada gambar dibawah.



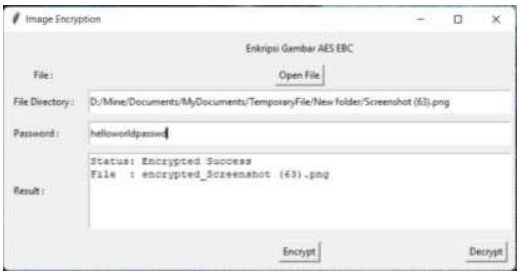
Gambar 1. Flowchart

Proses enkripsi dimulai dengan mengubah file dan kunci menjadi byte agar dapat diproses oleh algoritma AES. AES mengenkripsi setiap blok 128-bit dari file menjadi ciphertext berupa array byte terenkripsi, yang kemudian ditulis ke file dalam format ASCII. Untuk dekripsi, file hasil enkripsi diubah kembali menjadi byte, dan AES dengan mode ECB mendekripsi setiap 128-bit ciphertext hingga menghasilkan byte file asli, yang kemudian ditulis ke file.

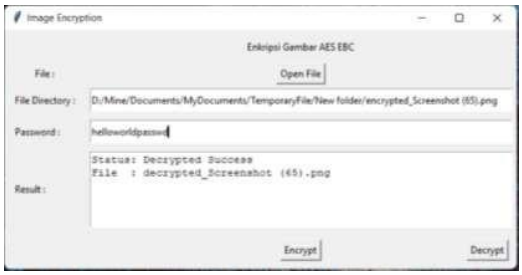
3. Result and Discussion

Artikel ini membahas implementasi algoritma AES untuk enkripsi dan dekripsi file menggunakan

Python dengan pustaka PyCryptodome. PyCryptodome menyediakan objek dan fungsi kriptografi, termasuk fungsi padding yang diperlukan dalam proses enkripsi dan dekripsi. Pustaka Tkinter digunakan untuk membuat antarmuka GUI berbasis objek. Enkripsi dilakukan menggunakan fungsi encrypt_data, yang mengubah kunci dan plaintext menjadi byte, kemudian mengenkripsi menggunakan objek AES dengan mode ECB. Hasil enkripsi berupa ciphertext dikembalikan sebagai string. Proses dekripsi menggunakan fungsi decrypt_data, yang menerima ciphertext dan kunci, mengubahnya ke byte, dan mendekripsi dengan objek AES menggunakan mode ECB. Fungsi ini mengembalikan plaintext dengan padding yang dihapus.



Gambar 2. Demo Enkripsi File



Gambar 3. Demo Dekripsi File

Gambar di atas menunjukkan cuplikan aplikasi enkripsi yang menerima input berupa direktori file dan password 16 karakter. Aplikasi menampilkan nama file hasil enkripsi di kolom "Result". Proses dekripsi dimulai dengan meminta direktori file terenkripsi dan password yang digunakan saat enkripsi. Implementasi algoritma AES 128 dengan mode ECB berhasil melakukan enkripsi dan dekripsi file, termasuk file audio, citra, dan video. Hasil enkripsi menunjukkan file tidak dapat dibuka, namun setelah dekripsi, file gambar dapat dibuka dengan baik.

Tabel 1. Hasil Enkripsi dan Dekripsi

Hasil Enkripsi	Hasil Dekripsi
	

3.1. PengujianSistem

Pada pengujian sistem, analisis pengujian aplikasi ini akan dilakukan pengujian proses penerapan algoritma AES dengan mde ECB pada enkripsi dan dekripsi *file* gambar dengan format *file* gambar yang berbeda, yaitu jpeg, jpg, bmp dan gif. Pengujian juga dilakukan dengan *file* gambar dengan *true colour* dan *greyscale* (tidak berwarna). Hal ini dilakukan untuk menguji bahwa proses enkripsi dan dekripsi menggunakan algoritma AES dengan mode ECB dapat dilakukan dengan berbagai format *file* gambar dan warna yang dimiliki *file* gambar tersebut. Pengujian ini juga dapat membuktikan bahwa algoritma AES dengan mode ECB dalam proses enkripsi dan dekripsi *file* gambar juga tidak menyebabkan perubahan terhadap ukuran, resolusi dan warna pada *file* gambar. Dengan kata lain yaitu aplikasi mampu mengamankan file gambar. Hasil pengujian ini dapat dilihat pada Tabel berikut

Tabel 2. Hasil Uji Enkripsi-Dekripsi AES terhadap Spesifikasi File Gambar

No.	Spesifikasi	Sebelum	Sesudah
1.	Format	jpeg	jpeg
	Warna	colour	colour
	Ukuran	1.48 Kb	1.48 Kb
	Resolusi	313 x 234 pixel	313 x 234 pixel
2.	Format	jpg	jpg
	Warna	colour	colour
	Ukuran	762 Kb	762 Kb
	Resolusi	1024 x 768 pixels	1024 x 768 pixels
3.	Format	bmp	jpg
	Warna	colour	colour
	Ukuran	12.4 Kb	12.Kb
	Resolusi	275 x 183 pixel	275 x 183 pixel
4.	Format	gif	jpeg
	Warna	grey	grey
	Ukuran	7.98 Kb	7.98 Kb
	Resolusi	225 x 225 pixel	225 x 225 pixel

4. Conclusion

Algoritma Advanced Encryption Standard (AES) dengan Mode ECB dapat diimplementasikan untuk melakukan enkripsi dan dekripsi file. Penggunaan AES akan mengenkripsi setiap 128bit blok dari file hingga menjadi sebuah ciphertext yang array dari byte yang terenkripsi. Proses dekripsi menggunakan algoritma AES dengan mode ECB akan mendekripsi tiap 128-bit dari ciphertext hingga menghasilkan byte array file asli. AES dengan mode ECB yang diimplementasikan pada bahasa pemrograman python dapat digunakan untuk mengenkripsi file gambar dengan format seperti jpeg, jpg, bmp, dan gif dengan baik, tanpa menyebabkan perubahan terhadap ukuran, resolusi dan warna pada file gambar. Dengan kata lain yaitu Algoritma Advanced Encryption Standard (AES) dengan mode ECB yang diimplementasikan pada Bahasa pemmrograman phyton mampu mengamankan file gambar dalam format jpeg, jpg, bmp, dan gif dengan baik.

Daftar Pustaka

[1] Chang, L & Rinaldi, M "Studi Dan Implementasi Advanced Encryption Standard Dengan Empat Mode Operasi Block Cipher"
[2] Fathurrozi Ahmad, Selviyani “Penerapan Algoritma Advanced Encryption Standard (AES-256) Dengan Mode CBC Dan Secure Hash Algorithm (SHA-256) Untuk Pengamanan Data File, Journal of Information adn Information Security (JIFORTY), Vol.2, No.2, 2021
[3] Simangunsong, P. B. N., & Fitri, K. (2018). Perancangan Aplikasi Pengamanan Citra Berwarna Dengan Algoritma RSA. Jurnal Teknik Informatika, 99-107
[4] Surian, D. (2006). Algoritma Kriptografi AES Rijndael. Jurnal Teknik Elektro, 97-101
[5] Wiguno, H. F. (2017). Aplikasi Pengamanan File Dan Pesan Teks Menggunakan AES 256 dan SHA 256 Berbasis Android.
[6] Basri (2015). Pendekatan Kriptografi Hybrid Pada Keamanan Dokumen Elektronik dan HypertextTransfer Protocol Secure (HTTPS) (Analisis Potensi Implementasi Pada Sistem Keamanan)