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, SE,Ak.,S.Si,M.Cs[2]

[a]Informatics Department, Udayana University
Jalan Raya Kampus Unud, Jimbaran, Bali, 80361, 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 are considered to be able to 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 SubBytes, the second step is ShiftRows, the third step is MixColums and the las step is AddRoundKey. 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*

Abstract

Pengamanan data merupakan salah satu hal penting untuk melindungi pesan dan informasi penting dari korupsi, kompromi atau kerugian supaya pesan dan informasi tersebut tetap aman. Teknik enkripsi dan dekripsi dinilai dapat mengamankan data dengan tepat dengan melindungi file agar tidak mudah untuk dibaca atau dilihat oleh pihak yang tidak berwenang. Pada penelitian ini penulis menggunakan algoritma kriptografi simetris *Advanced Encryption Standard* (AES) dan *Electronic Code Book* (ECB) sebagai solusi untuk masalah yang ada. Proses algoritma AES sendiri terbagi menjadi empat Langkah, Langkah pertama yaitu *SubBytes*, Langkah kedua yaitu *ShiftRows*, Langkah ketiga *MixColumns* dan Langkah terakhir yaitu *AddRoundKey*. Serta menggunakan algoritma ECB sebagai fungsi *hash*. Penerapan algoritma tersebut diterapkan ke dalam aplikasi *web* dan menggunakan *python* sebagai Bahasa pemrograman. 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 media seperti gambar, audio maupun video dengan tingkat keamanan yang baik.

Kata Kunci: Pengamanan Data, Enkripsi, Dekripsi, Algoritma AES-128, ECB

1. Introduction

Data yang bersifat pribadi menjadi objek yang disenangi oleh pihak-pihak yang tidak bertanggung jawab untuk dimanipulasi dan tidak digunakan sebagaimana mestinya. Oleh karena itu data yang bersifat

pribadi atau rahasia perlu dijaga keamanannya. Ada beberapa Teknik pengamanan data, diantaranya adalah Teknik enkripsi. Enkripsi merupakan sebuah proses penyandian pada pesan atau informasi dari yang semulanya bisa dimengerti menjadi sebuah pesan atau informasi yang sulit dimengerti hingga tidak dapat dimengerti lagi. Teknik enkripsi dapat mengamankan data karena data dapat diubah menjadi tidak terbaca sesuai dengan aslinya. Dan data yang telah terenkripsi dapat dibaca lagi setelah dilakukan proses dekripsi dengan menggunakan kunci yang tepat. Dan dengan mengenkripsi data file yang penting atau rahasia dapat meningkatkan keamanan data yang bersifat rahasia tersebut

Kriptografi merupakan studi bidang ilmu matematika yang mempunyai hubungan dengan aspek keamanan informasi seperti integritas data, keaslian entitas dan keaslian data (Ratno Prasetyo, 2016). Dalam ilmu kriptografi terdapat dua proses penyandian yang disebut enkripsi dan dekripsi. Enkripsi dilakukan pada proses pengiriman pesan atau informasi dengan cara mengubah data asli ke dalam bentuk kode kode yang menjadikannya rahasia, sedangkan dekripsi dilakukan pada proses penerimaan dengan cara mengubah data yang berisi kode kode rahasia tersebut bentuk data yang asli dan mudah dimengerti

Pada tanggal 29 September 2020 berita tentang *ransomware* yang melumpuhkan salah satu rumah sakit di Amerika Serikat, yang mengakibatkan data penting di rumah sakit habis terserang oleh sebuah virus, *ransomware* sendiri ialah salah satu jenis *malware* berbahaya yang menyerang system komputer untuk mengenkripsi file didalamnya. Maka dari itu diperlukan enkripsi file untuk file yang dianggap penting. Terdapat metode pada algoritma kriptografi yang cocok untuk memecahkan masalah pengamanan data, yaitu salah satunya adalah metode AES dan ECB. *Advanced Encryption Standard* (AES) adalah algoritma kriptografi simetris modern yang beroperasi dalam bentuk penyandian blok *(block cipher)* yang memproses blok data dengan ukuran 128-bit dengan panjang kunci 128-bit, 192-bit, atau 256-bit (Asep Suryana,2016). Terdapat beberapa metode dalam algoritma AES, diantaranya metode CBC, ECB, OFB, CTR, dan CFB untuk penyandian dengan metode block cipher.

2. Reseach Methods

2.1. AES

AES didesain berdasarkan jaringan substitusi–permutasi dan dapat dijalankan dengan efisien dalam perangkat lunak dan keras. AES berbeda dengan DES karena AES tidak menggunakan jaringan Feistel. AES adalah variasi dari Rijndael dengan ukuran blok tetap 128-bit dan ukuran kunci 128, 192, atau 256 bit. Sebaliknya, Rijndael sendiri didesain dengan ukuran blok dan kunci kelipatan 32-bit dengan minimum 128-bit dan maksimum 256 bit. Gambaran umum dari kerja algoritma AES adalah:

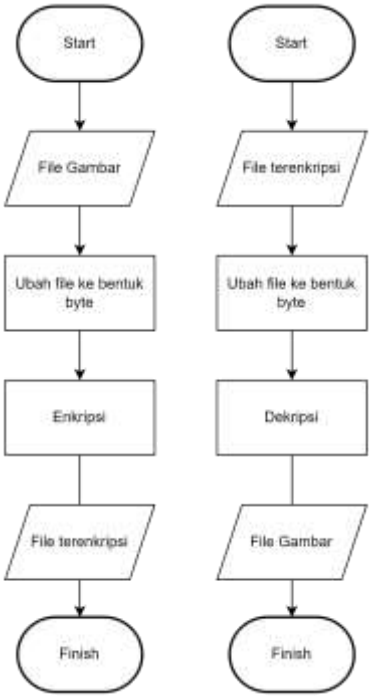
1. KeyExpansion, kunci ronde diturunkan dari kunci penyandian melalui penjadwalan kunci AES. AES membutuhkan kunci ronde 128-bit untuk tiap ronde ditambah satu.
2. Penambahan kunci ronde awalan:
    a. AddRoundKey, tiap bita digabung dengan satu bita dari kunci ronde dengan operasi XOR.
3. Selama 9, 11, atau 13 ronde:
    a. SubBytes, substitusi nonlinear yang tiap bita-nya ditukar dengan lainnya sesuai tabel acuan.
    b. ShiftRows, penukaran posisi yang tiga baris terakhirnya digeser beberapa kali.
    c. MixColumns, pencampuran linear yang bekerja pada tiap kolom "status", yaitu kombinasi keempat bita dalam tiap kolom.
    d. AddRoundKey
4. Ronde terakhir (ronde ke-10, 12, atau 14):
    a. SubBytes
    b. ShiftRows
    c. AddRoundKey

2.2. ECB

ECB *(Electronic Code Book)* adalah mode block cipher yang paling mudah berfungsi. Lebih mudah karena enkripsi langsung setiap blok input *plaintext* dan *output* berupa blok *ciphertext* terenkripsi. Umumnya, jika sebuah pesan berukuran lebih besar dari b bit, pesan tersebut dapat dipecah menjadi sekumpulan blok dan prosedur ini diulangi. Keuntungan menggunakan ECB adalah Enkripsi paralel blok bit dimungkinkan, sehingga merupakan cara enkripsi yang 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 2.1 Flowchart

Proses enkripsi dimulai dengan mengubah file dan kunci menjadi bentuk byte agar dapat dimasukkan ke fungsi enkripsi. Algoritma AES akan mengenkripsi setiap 128bit blok dari file hingga menjadi sebuah ciphertext yang array dari byte yang terenkripsi. Kemudian menulis hasil enkripsi ke sebuah file. Ciphertext yang ditulis pada file adalah ciphertext yang telah diubah dari bentuk byte ke ASCII. Sedangkan proses dekripsi dimulai dengan mengubah file hasil enkripsi menjadi bentuk byte. Proses dekripsi menggunakan algoritma AES dengan mode ECB akan mendekripsi tiap 128-bit dari *ciphertext* hingga menghasilkan *byte* file asli. Kemudian menulis *byte* hasil enkripsi ke dalam sebuah *file*.

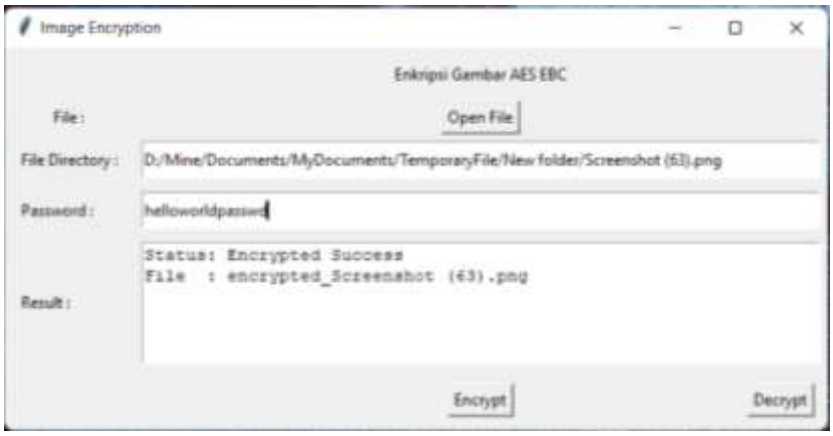
3. Result and Discussion

Pada artikel ini implementasi algoritma AES untuk melakukan enkripsi dan dekripsi *file* menggunakan bahasa pemrograman *python* dengan pustaka *pycryptodome* sebagai sumber kode dari algoritma AES yang digunakan. *Pycryptodome* digunakan karena pustaka tersebut berisi banyak objek dari algoritma kriptografi yang dapat digunakan dalam pengambangan aplikasi menggunakan bahasa pemrograman *python*. Tidak hanya objek dari algoritma kriptografi, pustaka tersebut juga berisikan fungsi yang dapat digunakan dalam proses enkripsi maupun dekripsi seperti fungsi *padding*. Pada aplikasi yang dibuat juga menggunakan pustaka tkinter yang merupakan pustaka GUI standar untuk Python. Tkinter menyediakan antarmuka berorientasi objek yang kuat ke toolkit GUI Tk.

Proses enkripsi menggunakan fungsi *encrypt*_data yang dibuat menggunakan objek AES dari *pycryptodome*. Fungsi tersebut menerima dua buah parameter yakni *plaintext* dan kunci yang memiliki panjang 16 karakter. Dalam fungsi *encrypt*_data data kunci dari parameter yang diterima diubah menjadi dalam tipe data *byte* menggunakan fungsi *encode* dengan bentuk utf-8 *encoding.* Kemudian data *plaintext* yang diterima ditambahkan *padding* dengan fungsi pad dengan *block size* 16. Setelah itu dibuat sebuah objek *cipher* dengan menggunakan objek AES dari *pycryptodome*, dengan memanggil fungsi new dari objek AES dengan parameter kunci yang telah diubah ke bentuk *byte* dan mode dari *block cipher* yaitu adalah mode ECB. Setelah objek *cipher* dibuat proses enkripsi dapat dilakukan dengan memanggil fungsi *encrypt* dari objek *cipher*. Fungsi *encrypt* akan mengembalikan *ciphertext* yang merupakan hasil dari fungsi

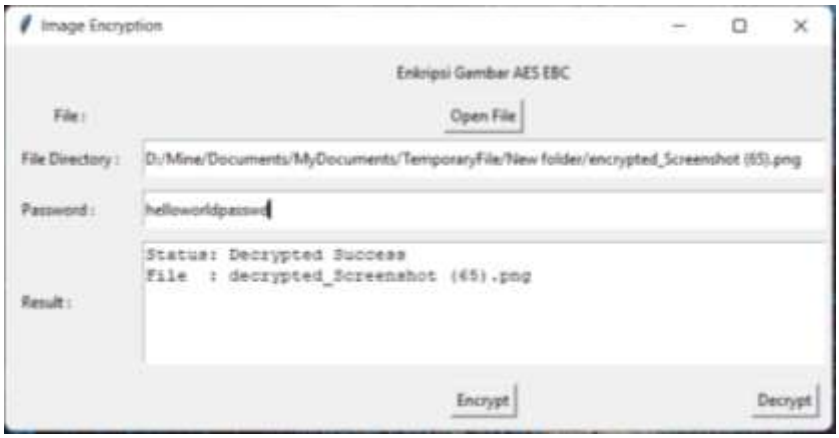
encrypt milik objek cipher. Fungsi encrypt_data akan mengembalikan ciphertext yang telah dikonversi dari bentuk *byte* ke *bentuk string*.

Sedangkan pada proses dekripsi dari aplikasi menggunakan fungsi *decrypt*_data yang dibuat menggunakan objek AES dari *pycryptodome*. Fungsi tersebut menerima dua buah parameter yakni *ciphertext* dan kunci yang memiliki panjang 16 karakter. Dalam fungsi *decrypt*_data data kunci dari parameter yang diterima diubah menjadi dalam tipe data *byte* menggunakan fungsi encode dengan. bentuk utf-8 encoding. Kemudian data *ciphertext* yang diterima juga akan diubah dalam bentuk byte menggunakan fungsi b64decode. Setelah itu dibuat sebuah objek *cipher* dengan menggunakan objek AES dari *pycryptodome*, dengan memanggil fungsi *new* dari objek AES dengan parameter kunci yang telah diubah ke bentuk *byte* dan mode dari *block cipher* yaitu adalah mode ECB. Setelah *objek cipher* dibuat proses dekripsi dapat dilakukan dengan memanggil fungsi *decrypt* dari objek *cipher*. fungsi *decrypt* akan mengembalikan *plaintext* yang merupakan hasil dari fungsi *decrypt* milik objek *cipher*. Fungsi decrypt_data akan mengembalikan *plaintext* dengan *padding* yang telah dihilangkan.



Gambar 3.1 Demo Enkripsi File

Gambar diatas dapat dilihat cuplikan dari aplikasi untuk proses enkripsi. Dalam aplikasi tersebut akan menerima input berupa direktori tempat file yang akan dienkripsi, password dengan panjang 16 karakter. Pada aplikasi juga terdapat kolom Result yang akan menampilkan nama file hasil enkripsi. Sedangkan untuk demo proses dekripsi dapat dilihat pada gambar dibawah. Pada proses dekripsi program akan meminta direktori file hasil enkripsi disimpan dan *password* yang digunakan pada proses enkripsi.



Gambar 3.2 Demo Dekripsi File

Implementasi dari algoritma AES 128 dengan mode block cipher ECB dapat melakukan enkripsi dan dekripsi file dengan baik. File audio, citra dan video dapat di enkripsi dan dekripsi dengan baik. Contoh enkripsi dan dekripsi dapat dilihat pada tabel dibawah. Hasil enkripsi dan dekripsi dapat dilihat pada tabel dibawah. Hasil enkripsi dari aplikasi menunjukan bahwa file yang dienkripsi tidak dapat dibuka. Kemudian setelah file didekripsi dapat dilihat pada tabel dibawah file gambar dapat dibuka dengan baik.

Tabel 3.1 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

No.	Hasil Uji Proses Enkripsi-Dekripsi		
	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

Tabel Hasil Uji Enkripsi-Dekripsi AES terhadap Spesifikasi File gambar

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.

References

[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)